

Magellan at NERSC

Progress Report for June 2010

R. Shane Canon
Lavanya Ramakrishnan
Jeff Broughton
Brent Draney
Keith Jackson

NERSC
Lawrence Berkeley National Laboratory

June 30, 2010



Executive Summary

The Magellan Project was funded by the American Recovery and Reinvestment Act to investigate the applicability of cloud computing for the Department of Energy's Office of Science (DOE-SC). Efforts range from evaluating what applications work well on today's commercial cloud offerings to seeing how jobs can be distributed across multiple DOE clouds to exploring emerging programming models like MapReduce. NERSC has adopted a data driven approach in understanding the motivations and requirements of scientific users and analyzing application performance in cloud like environments. NERSC has deployed a highly flexible testbed to support this research, allowing NERSC to explore a range of hardware and software configurations to provide insight into how these design points impact application performance, usability, and power efficiency.

This report covers the progress for the Magellan Project at NERSC since it began in September 2009 and focuses on the research aspects of the project. The initial half of the project was centered on procuring, installing, and deploying the hardware for the testbed. This work was completed in March 2010. The project has spent 57% of \$16,384,000 budget since June 30 which is inline with the project plan.

The Magellan testbed at NERSC includes computational, storage, and networking infrastructure. The Cluster consists of a 720 node cluster each with dual Intel Nehalem quad-core processors, 24 GB of RAM, and a QDR InfiniBand NIC. The deployed storage includes over a petabyte of disk storage, as well as archival storage. The system also includes network server nodes that will be connected to the 100Gb network planned for deployment by the DOE-SC funded Advanced Networking Initiative. The cluster uses xCAT and the Moab Adaptive Computing Suite to provide advanced provisioning and scheduling capabilities. This software combination will allow NERSC to explore different models of providing access to computing resources including traditional batch scheduling, dynamic image provisioning and virtual private clusters.

While the testbed has been in operation for only a few months, NERSC has made significant progress in addressing many of the research questions. Early on NERSC solicited information from users expressing an interest in Magellan. The questionnaire included questions both about motivations for using Cloud computing as well as details about their application and data requirements. The results of the survey indicate that many of the users are interested in Cloud computing as a way to access additional computing cycles. However, users are also drawn to the ability to customize and share software environments for their applications. The survey also highlighted that users are still looking for capabilities like parallel file systems and archival storage found in centers like NERSC. The feedback from this survey guides the project in deciding what capabilities users are interested in exploring.

NERSC has also conducted several benchmarking activities to study how applications perform in cloud environments versus traditional batch clusters. The benchmarks include the NERSC6 benchmarks, is representative of the workloads observed amongst NERSC users, and the HPCC benchmark suite. These benchmarks were run on NERSC clusters and supercomputers, virtual machines on Magellan, and a commercial cloud system. The results of these benchmarks highlight the importance of low-latency, high-bandwidth interconnects. Applications performed 2x to 50x slower on the commodity networks running on virtual machines. Applications with the most communications and synchronization experienced the most significant impact. On the other hand, NERSC confirmed that serial, high-throughput oriented applications like BLAST performed well on both commercial clouds and in frameworks like Hadoop and Microsoft's Azure.

The Magellan Project has already made an impact for users. A facility problem at the Joint Genome Institute led to a pressing need for backup computing hardware to maintain their production sequencing operations. NERSC in partnership with ESNet was able to provision Magellan hardware in a Hardware as a Service (HaaS) model to help them meet their demands. This experience demonstrated the potential for Cloud computing but also highlights areas where more integration is required. NERSC users also have access to the Magellan testbed to run in a standard batch cluster environment. The users have used nearly 13 million core hours since the system completed acceptance in March. Users from all the DOE-SC offices have taken advantage of the system.

Magellan Progress Report

NERSC's experience to date on using Magellan and Cloud oriented technologies have exposed several areas that require further attention. Challenges include: securing and protecting cloud resources both from inside and outside threats; providing access to critical shared resource like global file systems; leveraging high-performance interconnects like InfiniBand in virtualized environments; and porting scientific applications to cloud environments and cloud programming models. Addressing these challenges will require a combination of technologies, system integration, and policy.

During the remainder of the Magellan Project, NERSC will continue to address research questions and explore approaches to addressing some of the challenges. The efforts to date clearly show both the promise of cloud computing, as well as the areas that require more effort and investment.

Contents

1	Project Objectives	5
2	Background	7
2.1	Service Models	7
2.1.1	Infrastructure as a Service	7
2.1.2	Platform as a Service	8
2.1.3	Software as a Service	8
2.2	Magellan Project Activities	8
3	Project Activities	11
3.1	User Survey	11
3.1.1	Overview	12
3.1.2	Application Characteristics	12
3.1.3	Application Case Studies	15
3.1.4	Summary of User Survey	16
3.2	Deployment	17
3.2.1	Testbed	17
3.2.2	Software Stack	19
3.3	Benchmarking	20
3.3.1	Experiment Setup	21
3.3.2	Evaluation of Performance of Commercial Cloud Platform	21
3.3.3	Comparison of Virtualization and Bare Metal Performance on Magellan	22
3.3.4	HPCC	22
3.3.5	Comparison of Performance of BLAST	24
3.3.6	Hadoop Benchmarks on Magellan	24
3.3.7	Summary of Benchmarking	26
3.4	Flash Storage Evaluation	26
3.5	Usage and Impact	26
3.5.1	Hardware as a Service – Joint Genome Institute	28
3.5.2	Traditional Cluster Usage	28
4	Challenges and Future Plans	30
4.1	Challenges	30
4.1.1	Security and User-Provided Images	30
4.1.2	InfiniBand Support for Virtual Machines	31
4.1.3	Secure Access to Shared File Systems from Virtual Machines	32
4.1.4	Hadoop Programming Interface	32
4.1.5	Eucalyptus Accounting and Scheduling	32
4.2	Future Plans	33

5	Conclusions	35
A	User Requirements Detail	38
A.1	Questionnaire	38
A.2	List of Projects expressing interest	42
B	Relevant Publications	53

Chapter 1

Project Objectives

The Magellan Project was funded by the American Recovery and Reinvestment Act to investigate the applicability of cloud computing for the Department of Energy’s Office of Science (DOE-SC). Cloud computing has recently emerged as a new model for providing computing infrastructure and services. While cloud computing has similarities with other distributed computing models such as Grid and Utility computing, cloud computing has leveraged virtualization and a *pay-as-you-go* business model that has made it attractive to the enterprise computing space. Cloud computing has been particularly successful with customers lacking significant IT infrastructure or customers that have quickly outgrown their existing capacity. This new model has also attracted the interest of the scientific computing community. Many communities are rapidly outgrowing their own infrastructure or periodically require access to surge resources. In many cases deploying new hardware requires significant additional space, facilities investment and trained personnel. In some cases, expanding facilities is not feasible. Cloud computing can potentially help address these requirements. The Magellan Project at NERSC is focused on understanding the unique requirements of DOE SC applications and the role cloud computing can play in scientific communities.

One challenge with exploiting cloud computing for science is that the scientific community has needs that can be significantly different from typical enterprise customers. Applications often run in a tightly coupled manner at scales greater than most enterprise applications require. This often leads to bandwidth and latency requirements that are more demanding than most cloud customers. Scientific applications also typically require access to large amounts of data including large volumes of historical data. This can lead to large startup cost and data storage cost. Magellan at NERSC will analyze these requirements using an advanced testbed, a flexible software stack, and performing a range of data gathering efforts.

In order to assess the effectiveness of cloud computing for DOE-SC, the Magellan project will address several key questions.

1. What part of the DOE SC workload can be supported within an existing commercial cloud model, and which cloud model is appropriate for which applications: Hardware as a Service (HaaS), i.e., operating system virtualization as in Amazon’s EC2 cloud; Parallel Frameworks, e.g., Google’s MapReduce cloud; Software as a Service, as in Microsoft’s cloud service; or Data as a Service, as in Amazon’s S3 cloud?
2. What is the efficiency of computing in the cloud and how does that depend on workload characteristics such as communication volume, message size, synchronization frequency, I/O bandwidth requirements, and programming models? How do potential underlying hardware options compare to one another and to traditional high-bandwidth, low-latency clusters?
3. How cost-effective are commercial clouds relative to privately owned clouds for various workloads? How cost-effective (in terms of total cost of ownership) are clouds relative to DOE’s current ad hoc approach to mid-range computing?

4. Are there changes to commercial clouds that would make their offerings more appropriate for mid-range scientific computing? Are commercial clouds a viable option when it comes to addressing sharp spikes in the workload of scientific applications?

Another objective of the Magellan Project is to explore how new models of distributed computing may be applied to scientific computing. These new models include evaluating existing frameworks like MapReduce and Software as a Service, and answering the following questions.

1. Can a single DOE front-end to a cloud provide a convenient mechanism for accessing both private DOE clouds and commercial clouds? How would jobs need to be packaged and identified to make them location-independent?
2. Can some popular mid-range applications be provided as an application service within the cloud, thereby making scientists even more productive than they are today?
3. Can multiple DOE clouds (at NERSC and ALCF) provide high availability by allowing job queues at one site to be redirected to another due to planned or unplanned outages? Can they also provide load balancing across the two clouds?
4. Can multiple DOE clouds run at different security levels and allow jobs with different security requirements to exist within a single logical cloud?

In order to successfully address the research agenda of the Magellan Project, NERSC has adopted a **data-driven approach** and deployed a testbed that is **flexible**. NERSC is focusing on understanding the requirements of scientific users by soliciting data via a detailed questionnaire. This data-driven approach is continued in the benchmarking and performance analysis efforts. Future efforts will include assessing the energy efficiency and cost effectiveness of cloud computing for scientific applications. Both the hardware architecture and software stack have been chosen to allow NERSC to explore a variety of models with the testbed. For example, the system provides both a high-bandwidth, low-latency quad-data rate InfiniBand network as well as a commodity Gigabit Ethernet network. While InfiniBand may be unusual in a typical commercial cloud offering, it allows NERSC to investigate a range of performance points and measure the impact on application performance.

Chapter 2

Background

The term “cloud computing” covers a range of delivery and service models. The common characteristic of these service models is an emphasis on *pay-as-you-go* and elasticity, the ability to quickly expand and collapse the utilized service as demand requires. Thus new approaches to distributed computing and data analysis have also emerged in conjunction with the growth of cloud computing. These include models like MapReduce and scalable key-value stores like Big Table [4].

Cloud computing technologies and service models are attractive to scientific computing users due to the ability to get on-demand access to resources to replace or supplement existing systems, as well as the ability to control software environment. Scientific computing users and resource providers servicing these users are considering the impact of these models and technologies. In this section, we briefly describe the cloud service models and technologies to provide some foundation for the discussion.

2.1 Service Models

Cloud offerings are typically categorized as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each of these models can play a role in scientific computing.

The distinction between the service models is based on the layer at which the service is abstracted to the end user (e.g., hardware, system software, etc.). The end user then has complete control over the software stack above the abstracted level. Thus in IaaS a virtual machine or hardware is provided to the end user and the user then controls the operating system and the entire software stack. We describe each of these service models and visit existing examples in the commercial cloud space to understand their characteristics.

2.1.1 Infrastructure as a Service

In the Infrastructure as a Service provisioning model, an organization outsources equipment including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. In the commercial space, the client typically pays on a per-use basis for use of the equipment.

Amazon Web Services is the most widely used IaaS cloud computing platform today. Amazon provides a number of different levels of computational power for different pricing. The primary methods for data storage in Amazon EC2 are S3 and Elastic Block Storage (EBS). S3 is a highly scalable key-based storage system that transparently handles fault tolerance and data integrity. EBS provides a virtual storage device that can be associated with an Elastic Computing instance. S3 charges for space used per month, the volume of data transferred, and the number of metadata operations (in allotments of 1000). EBS charges for data stored per month. For both S3 and EBS, there is no charge for data transferred to and from EC2 within a domain (e.g., the U.S. or Europe).

Eucalyptus is open source software that provides a private cloud IaaS service. Eucalyptus services include a cloud controller, a cluster controller, and node controllers for managing the virtual machines (VMs). The interface to launch and manage VMs is compatible with Amazon EC2. In addition, Walrus and the storage controller provide persistent storage and block store services that are interface compatible with Amazon S3 and Elastic Block Storage.

In Magellan, in conjunction with other synergistic activities we use Amazon EC2 as the commercial cloud platform to understand and compare an existing cloud platform. We use Eucalyptus to set up a private cloud IaaS platform on Magellan hardware for detailed experimentation on providing cloud environments for scientific workloads.

2.1.2 Platform as a Service

Platform as a Service (PaaS) provides a computing platform as a service, supporting the complete life cycle of building and delivering applications. PaaS often include facilities for application design, development, deployment and testing, and interfaces to manage security, scalability, storage, state, etc. Windows Azure, Hadoop and Google App Engine are popular PaaS offerings in the commercial space.

Windows Azure is Microsoft's offering of a cloud services operating system. Azure provides a development, service hosting and service management environment. Windows Azure provides an on-demand compute and storage resources for hosting applications to scale costs. The Windows Azure platform supports two primary virtual machine instance types – the Web role instances and the Worker role instances. It also provides Blobs as a simple way to store data and access it from a virtual machine instances. Queues provide a way for Worker role instances to access the work quantum from the Web role instance. While the Azure platform is primarily designed for web applications, its use for scientific applications is being explored [8, 9].

Hadoop is an open-source software that provides capabilities to harness commodity clusters for distributed processing of large data sets through the MapReduce [5] model. The Hadoop streaming model allows one to create map and reduce jobs with any executable or script as the mapper and/or the reducer. This is the most suitable model for scientific applications that have years of code in place capturing complex scientific processes.

The Hadoop File System (HDFS) is the primary storage model used in Hadoop. HDFS is modeled after the Google File system and has several features that are specifically suited to Hadoop/MapReduce. Those features include exposing data locality and data replication. Data locality is a key aspect of how Hadoop achieves good scaling and performance: Hadoop attempts to locate computation close to the data. This is especially true in the map phase, which is often the most I/O intensive phase.

Hadoop provides a platform for managing loosely coupled data-intensive applications. In Magellan synergistic activities, we used the Yahoo! M45 Hadoop platform to benchmark BLAST. In addition, Hadoop has been deployed on Magellan hardware to enable our users to experiment with the platform.

2.1.3 Software as a Service

Software as a Service provides access to an application or software that has a specific function to the end user. Amazon provides a higher-level service such as Elastic Map Reduce that allows applications to run MapReduce jobs on EC2. In our project activities, we use the Windows Azure BLAST service to run BLAST jobs on the Windows Azure platform. Similar technologies such as HBase provide specialized functions on top of Hadoop. HBase is being used as a platform for managing genomics data and is being tested on Magellan hardware. HBase provides key value access and tabular storage and provides the potential for addressing these problems that cannot be solved with conventional relational database management systems.

2.2 Magellan Project Activities

Magellan project activities have been guided by two basic principles: *data-driven* and *flexibility*. Our efforts are centered around collecting data about user needs and performance of cloud applications from a range of

Table 2.1: Magellan Project Activities in reporting period

Activity	Begin and End Date
Installation & Acceptance	Dec 2009 - March 2010
User Survey	Nov 2009 - Feb 2010
Benchmarking of commercial cloud platforms	March - April 2010
Benchmarking of BLAST on commercial platforms	Oct - Nov 2009
Hardware and software deployment	Jan - Mar 2010
Hadoop benchmarking	Apr 2010 -
Baseline benchmarking of existing private cloud platforms	May - June 2010
Flash storage evaluation	June 2010
Hardware as a service/JGI	March - May 2010
Eucalyptus testing	Dec 2009 - Feb 2010 and June 2010

sources – benchmarks and user applications on both private and cloud platforms. This data-driven approach helps us understand the suitability of cloud environments for science. The diverse needs of scientific users also drive us towards a flexible software stack and approach in our project activities.

Table 2.1 summarizes the project activities performed in the reporting period. Our project activities included a detailed user survey to understand the applications and needs of our users, benchmarking efforts to understand the performance of cloud environments, evaluation of various technologies including flash storage, Hadoop, Eucalyptus, and deployment of hardware and software for wider use.

Collaborative and Synergistic Activities The Magellan project has a number of synergistic activities through its collaboration. We outline these activities below:

- The benchmarking of commercial cloud platforms was performed in collaboration with the IT department at Lawrence Berkeley National Laboratory (LBNL), which manages some of the mid- range computing clusters for scientific users; the Advanced Technology Group at NERSC, which understands the requirements of current and emerging NERSC applications to find hardware design choices and programming models; and the Advanced Computing for Science (ACS) Department in the Computational Research Division (CRD) at LBNL, which seeks to create software and tools to enable science on diverse resource platforms. Access to commercial cloud platforms was also possible through collaboration with the IT division and the University of California Center for Information Technology Research in the Interest of Society (CITRIS), which had existing contracts with different providers.
- We collaborate closely with the Argonne Magellan group through bi-weekly teleconferences and periodic meetings. Project personnel also have friendly early access to test environments at both sites. Early efforts are underway to run applications across the two sites. Argonne’s MG-RAST, a fully automated service for annotating metagenome samples, and the Joint Genome Institute’s MGM pipeline are being tested across both sites with fail-over fault tolerance and load balancing.
- We leveraged the partnership between the Joint Genome Institute (JGI) and NERSC to benchmark the IMG and MGM pipelines on a variety of platforms. Project personnel were also involved in pilot projects for the Systems Biology Knowledge Base. They provided expertise with technologies such as HBASE which were useful in guiding the deployments on Magellan.
- Magellan project personnel were part of the team of researchers from LBNL who received the Best Paper Award at ScienceCloud 2010. The paper describes the feasibility of porting the Nearby Supernova Factory pipeline to the Amazon Web Services environment and offers detailed performance results and lessons learned from various design options. The Nearby Supernova Factory is a Magellan application.
- The Advanced Networking Initiative (ANI) is another ARRA funded ASCR project. The goal of ANI is to help advance the availability of 100Gb networking technology. The project has three major sub-projects: deployment of a 100Gb prototype network, the creation of a network testbed, and ANI

Magellan Progress Report

research projects. The later will attempt to utilize the network and testbed to demonstrate how 100Gb networking technology can help advance scientific projects in areas such as climate research and High-Energy Physics. The Magellan Projects at NERSC and ANL will help support ANI by providing critical end points on the network to act as data producers and consumers. ANI research projects such as Climate 100 and OSG are already working on Magellan to prepare for future demonstrations.

Chapter 3

Project Activities

In this section we will describe the efforts to date and summarize the conclusions made so far. The efforts thus far have included: gathering requirements from potential cloud computing users; hardware and software deployment; initial benchmarking; and flash storage evaluation.

3.1 User Survey

Cloud computing introduces a new usage or business model and additional new technologies that have previously not been applied at a large scale in scientific environments. The virtualization technology that enables the cloud computing business model to succeed for Web 2.0 applications can be used to configure privately owned virtual clusters that science users can manage and control. In addition, cloud computing introduces a myriad of new technologies for resource management at sites and programming models and tools at the user level.

It is critical to understand the needs of scientific applications and users and analyze these requirements in concert with existing cloud computing platforms and solutions. In addition to the traditional NERSC users, we identified three categories of scientific community users that might benefit from cloud computing resources at NERSC:

Private cluster. Some scientific users prefer to run their own private clusters for a number of reasons. They often don't need the concurrency levels achievable at supercomputing centers, but do require guaranteed access to resources for specific periods of time.

Personalized environment. A number of scientific applications have strong OS version dependencies and need environments that are consistent with local cluster or desktop environments. In these cases, while users might not care where the resources are located, they desire the flexibility associated with custom software environments.

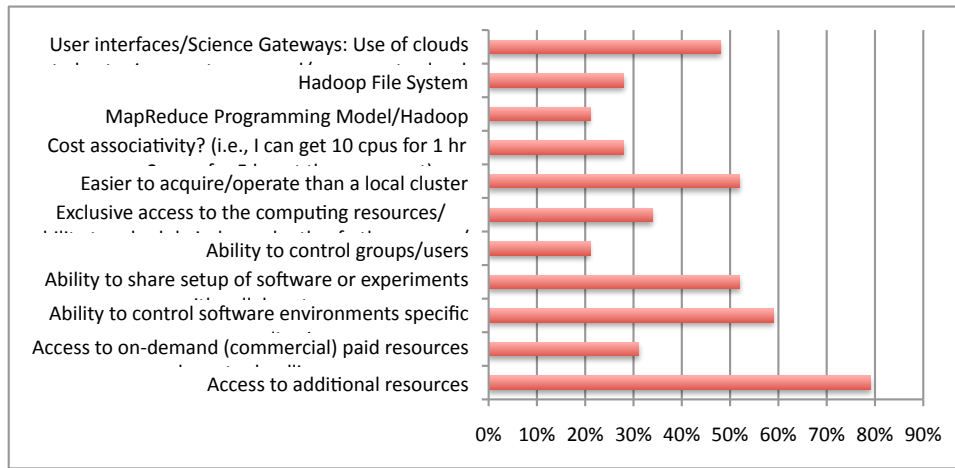
Science Gateways. Users of well-defined computational workflows often prefer to have simple web-based interfaces to their application workflow and data archives. Web interfaces enable easier access to resources by non-experts, and enable wider availability of scientific data for communities of users in a common application area (e.g., Virtual Organizations).

We conducted a survey to understand the requirements and expectations of the user community. We requested NERSC users and other communities in DOE that were interested in cloud computing to detail their application characteristics and expectations from cloud computing. The survey is available through the NERSC Magellan website at <http://magellan.nersc.gov/>

Table 3.1 shows the DOE Program Offices that the survey respondents belong to. The survey form and the list of respondents are included in the Appendix. We summarize the details and the results from the survey below.

Table 3.1: Percentage of survey respondents by DOE Office

Advanced Scientific Computing Research	17%
Biological and Environmental Research	9%
Basic Energy Sciences - Chemical Sciences	10%
Fusion Energy Sciences	10%
High Energy Physics	20%
Nuclear Physics	13%
Advanced Networking Initiative (ANI) Project	3%
Other	14%



(a) Features of cloud computing attractive to scientific users

Figure 3.1: Understanding the features of cloud computing that is of interest to scientific users.

3.1.1 Overview

We asked our users to identify the features of cloud computing that were most attractive to them. Users were given several options and were allowed to select more than one category. Figure 3.1a shows the responses selected by our users. Access to additional resources was the most common motivation for most of our users and was selected by 79% of our respondents. A large number of users also wanted the ability to control software environments (59%) and share software or experiment setup with collaborators (52%), which is hard to do in today's supercomputing setup. Additionally, clouds were also attractive due to ease of operation compared to a local cluster (52%) and access to end users of science gateways (48%). A fraction of the users were interested in exploring the use of Hadoop and the MapReduce programming model and the Hadoop File System for their science problems.

In addition, 90% of our survey respondents also mentioned that there were others in their scientific community that were either investigating or interested in investigating cloud computing. A majority of the users also said other collaborators would be able to use their cloud setup.

3.1.2 Application Characteristics

Cloud computing provides a fundamentally different model of resource access than supercomputing centers today. As we explore the applications that might benefit from the cloud environment and consider design decisions, we need to understand the application characteristics better. We asked our users a number

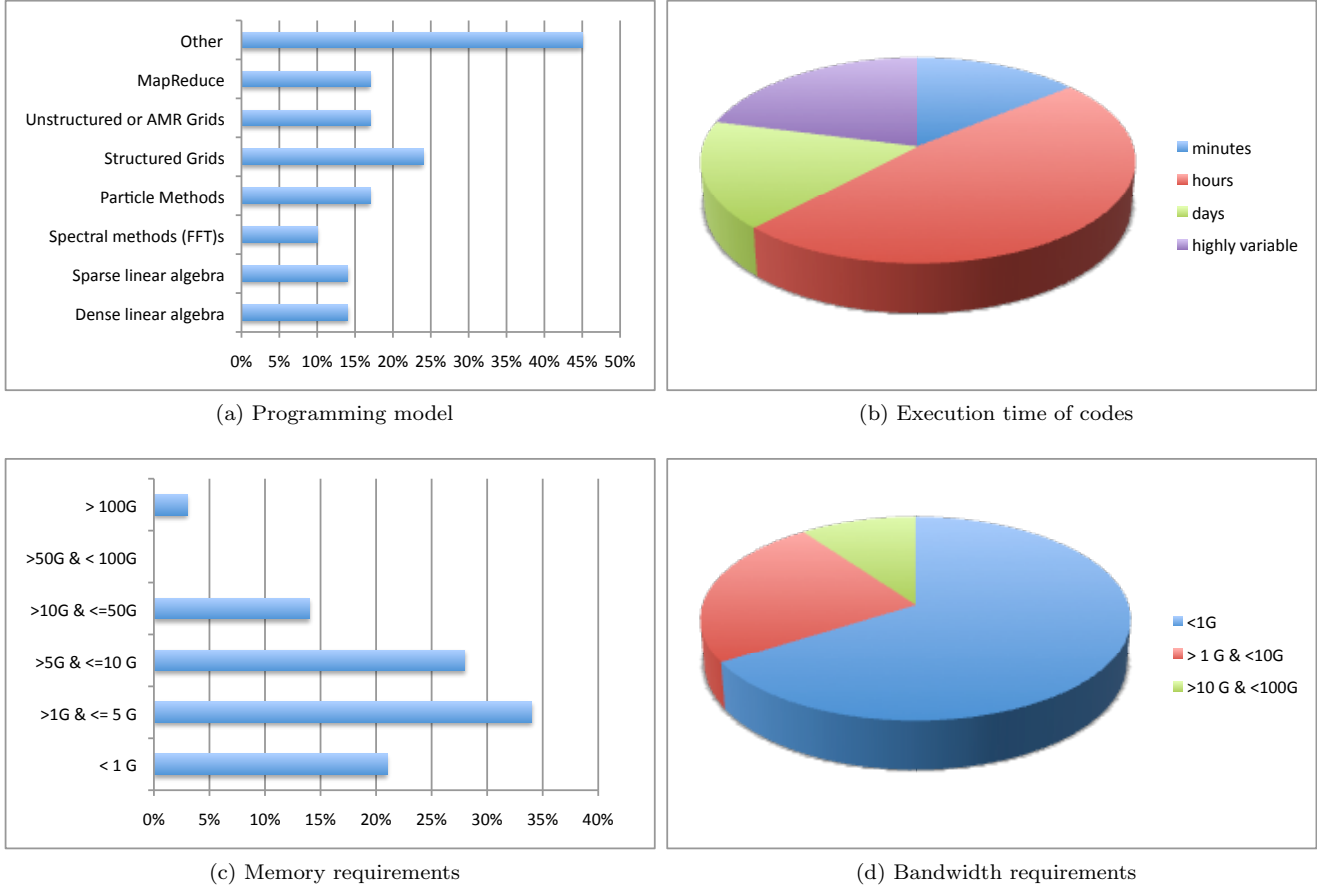
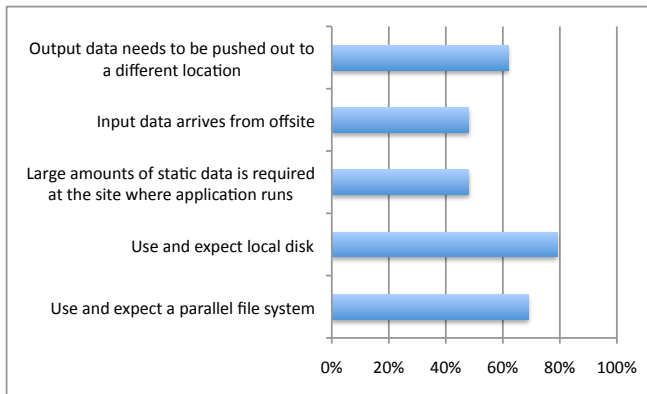


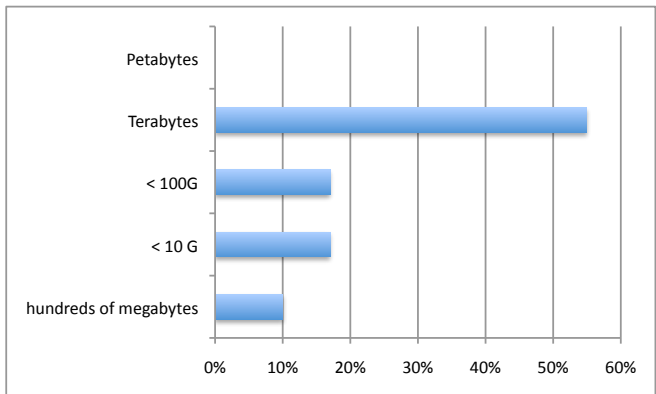
Figure 3.2: Understanding the application characteristics of codes of users that are interested in cloud computing as a platform

of questions about their applications. The users interested in using cloud computing had a varied set of programming models, from traditional supercomputing parallel models, to MapReduce, to custom codes that used one or more of these programming models. Figure 3.2b shows that the application's execution time for a majority of our users was in the order of hours (48%) or minutes (14%). In addition, the memory footprint of a large number of these applications is less than 10 GB, and the bandwidth requirement is 1 Gbps today.

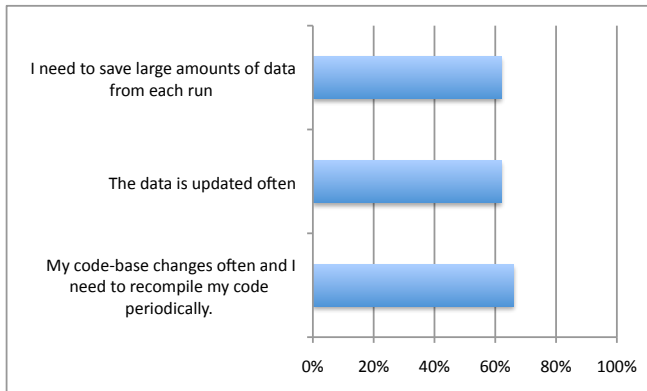
Similarly we polled our users on their data needs from the cloud. A large number of the scientific applications rely on a parallel file system and expect local disk on the nodes. Applications also have large amounts of data (order of gigabytes or terabytes) that they assume are available at the sites and/or data that arrives from remote sites. Virtualization is expected to have an impact on application performance. While a number of users anticipated their codes would be affected, the other advantages of cloud computing still made it an attractive platform for them. Scientific applications additionally face other challenges when running in cloud environments. Code bases often change and their codes need to be recompiled periodically, requiring new virtual images to be created. Additionally, data is updated often and/or large data from the runs need to be saved since virtual machines do not maintain persistent state.



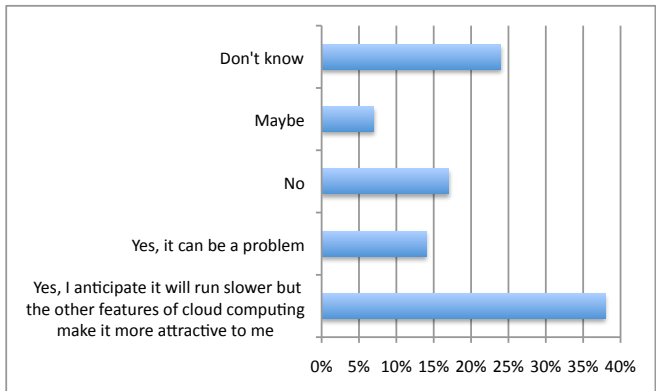
(a) Data characteristics of application



(b) Persistent disk requirements



(c) Characteristics that impact image creation



(d) Performance impact of cloud environments

Figure 3.3: Understanding application characteristics that might impact design decisions in cloud environments

3.1.3 Application Case Studies

We followed up the survey with one-on-one discussions with some of the application groups. We summarize the discussions with some of the early cloud adoptors and their requirements from NERSC Magellan.

Climate 100. Climate scientists are better able to understand what is happening and evaluate the effectiveness of possible mitigations of global climate change by generating and sharing increasingly large amounts of data. The goal of the Climate 100 project is to bring together middleware and network researchers to develop the needed tools and techniques for moving unprecedented amounts of data using the ANI 100 Gbps network. The Climate 100 data consists of on the order of a million files that average about 100 megabytes each. Climate 100 could benefit from cloud environments such as virtual machines and Hadoop to perform large-scale data analysis on the climate data. The volume of data requires coordination of network, compute and disk resources.

Open Science Grid/STAR. The STAR nuclear physics experiment studies fundamental properties of nuclear matter from the data collected at Brookhaven National Laboratory’s Relativistic Heavy Ion Collider. The STAR experiment has access to a number of different resource sites that are used for regularly processing experiment data.

STAR experiments are embarrassingly parallel applications (i.e., non-MPI codes) where each job fits in one processor or core. The STAR suite of experiments consists of various analysis and simulation programs. The Monte Carlo cases in STAR are well suited to run in cloud environments due to the minimal requirements on data transfer and I/O. The ability to control the software stack in a virtual machine is very attractive to the community due to the complexity of the codes. The ability to grow in scale during burst periods using existing virtual machine images would greatly enhance the scientific experimentation. The group has previously demonstrated the use of Amazon EC2 resources to process large amounts of data in time for the Quark Matter physics conference. Different groups in the STAR community are interested in cloud computing and virtual machines. The community is investigating use of virtual machine images as a way of packaging software for future experiments.

Supernova Factory. The Supernova Factory project is building tools to measure the expansion of the universe and energy. The experiment has a large number of simulations that require custom environments where the end results also need to be shared with other collaborators, and hence there is a need to co-allocate compute and network resources and move the data to storage resources. The Supernova Factory relies on large data volumes for the supernova search, and the code base consists of a large number of custom modules. The complexity of the pipeline makes it necessary to have specific library and OS versions and ends up being a barrier to making use of other large resource. The Supernova Factory project finds cloud computing attractive due to the ability to control software environments and the ability to manage and control user accounts and groups for access to the software. Initial experiments conducted by the group in collaboration with Magellan project personnel on Amazon EC2 show that the cloud is a feasible platform for this application. There is also interest in using Hadoop to coordinate and manage the loosely coupled jobs.

ATLAS. The ATLAS project is investigating the use of cloud platforms to support analysis jobs. The ATLAS project has hundreds of jobs that operate on terabytes of data and can greatly benefit from timely access to cloud resources. The cloud environment also promises to be an effective platform for transitioning scientific codes from testing on the desktop to large-scale cloud resources. The group is investigating the use of virtual machine images for distribution of all required software [3]. This would enable sites to boot the virtual machines at different sites with minimal or no work involved with software management.

Integrated Microbial Genomes (IMG) Pipeline. The Integrated Microbial Genomes (IMG) pipeline at the DOE Joint Genome Institute (JGI) provides analysis of microbial community metagenomes in the integrated context of all public reference isolate microbial genomes. IMG has workloads that need to run periodically every few weeks to months for content maintenance [2]. Timeliness of completion of workloads is critical for the community, and the tremendous growth of these data sets makes access to large number of resources critical. The computational stage consists of functional annotation of individual genes, identification of pair-wise genes, and identification of chromosomal clusters. The most computationally intensive step

is performing a BLAST analysis against a reference database. Subsequent steps characterize the genes based on the alignments reported by BLAST. The BLAST output alone is typically over a terabyte. Consequently the analysis of the output to find the top matches and identify taxons can be time consuming and must be done in parallel. There is an interest to use technologies such as Hadoop to ease management of these loosely coupled application runs.

In early experiments conducted by the Magellan team personnel, we benchmarked BLAST on a variety of HPC and cloud platforms. These results are discussed in greater detail in the benchmarking section.

Currently the output of all vs. all pairwise gene sequence comparisons is stored in compressed files. However modifying individual entries and querying the data is not easy in this format. The group is interested in exploring the use of HBase for managing data that will allow the users to update individual rows and perform simple queries.

3.1.4 Summary of User Survey

The results of the detailed survey have helped us in understanding the science requirements for cloud environments and have influenced the direction of research in the project. We summarize the requirements gathered from the user survey and corresponding project activities:

- The user requirements for cloud computing are diverse, ranging from access to custom environments to the MapReduce programming model. These diverse requirements guided our *flexible* software stack. Users of Magellan at NERSC will have access to a) traditional batch queue access with the additional capability of custom software environments through xCAT; b) customized virtual machines through Eucalyptus frontend enabling users to port between commercial providers and the private cloud; c) a Hadoop installation that allows users to access the MapReduce programming model, the Hadoop Distributed File System and other job management features such as fault tolerance. More details of the software stack are presented in Section 3.2.2.
- It is important to understand if commercial cloud platforms such as Amazon EC2 and private cloud software such as Eucalyptus and Hadoop met the needs of the science. We identified the existing gaps, which are summarized in the position paper *Defining Future Platform Requirements for e-Science Clouds* (available in the Appendix).
- In addition to understanding the gaps, we undertook a benchmarking effort to understand the computational scalability of commercial cloud platforms. These results are presented in Section 3.3.
- Our flexible software stack enables us to serve diverse user groups. The resulting mixed-use cluster needs tools that support reconfiguration of individual nodes into any of the cluster types. We also need the ability to manage accounting, data management, security, etc. across these software stacks. Some of the challenges and future plans are detailed in Sections 4.1.4 and 4.1.5.
- Our users were also interested in a private cloud that would enable them to get the benefits of cloud environments in conjunction with other facilities provided at supercomputing centers. For example, HPC systems typically provide high-performance parallel file systems that enable parallel coordinated writes to a shared filesystem with high bandwidth and capacity. HPC centers also typically provide an archival storage system to archive critical output and results.
- Cloud computing addresses the portability of the software stack, a known issue with current grid systems. In current supercomputing centers, the sites manage the operating system and common middleware that is needed across multiple groups. Users compile and install their applications on specific systems (often with help from site personnel for optimizations necessary for particular hardware). As we move to the cloud computing model, sites need to be able to provide tooling and support for managing a diverse set of kernels and operating systems that might be required by specific groups. The

clear separation of responsibilities for software upgrades and operating system patches no longer exists, and sites will need mechanisms to bridge the gap between supporting user-supported images and site security policies. A cloud system where users have complete control of their images, has certain implications on site security policies. Our security efforts are guided by these custom environments and are discussed in Section 4.1.1.

3.2 Deployment

3.2.1 Testbed

A dedicated testbed was deployed to support the Magellan research agenda. Deploying a testbed versus acquiring services on existing commercial cloud systems was critical to allowing us to address the research questions. The flexibility to adjust aspects of the system software and hardware are necessary to understanding how these design points impact application performance and usability. For example, the ability to measure the impact of virtualization or high-speed interconnects will provide insight for potential commercial or private cloud offerings.

Computational Hardware

The Magellan cluster uses IBM's iDataplex system, a hardware platform specifically designed for large-scale systems with a focus on energy efficiency. The system consists of nine iDataplex racks, each containing 80 nodes. A summary of the configuration for each node is shown in Table 3.2. The total system has over 60 TFLOPS of peak floating pointing performance. The system provides a high-performance InfiniBand network which is often used in HPC oriented clusters, but is not yet common in mainstream commercial cloud systems. Since the network has such a large influence on the performance of many HPC applications, the ability to explore the range of networking options from native InfiniBand to virtualized Ethernet was an important design goal.

A production cluster, Carver, was deployed in conjunction with the Magellan cluster. Both systems were procured under the same Request for Proposal process. The systems use identical hardware configurations. Some of the benchmarking results have been collected on the Carver system, but the results should be identical for the Magellan hardware.

Table 3.2: Node configuration on the Magellan cluster. There are 720 nodes in the Magellan cluster.

Feature	Description
Processor	Dual 2.66 GHz Intel Quad-core Nehalem
Memory	24 GB (48 GB in 160 nodes)
Local Disk	1 TB SATA (in 160 nodes)
High Performance Network	40 Gb InfiniBand (4X QDR)
Ethernet Network	On-board 1 Gb
Management	IPMI

The InfiniBand fabric was built using InfiniBand switches from Voltaire. Each node link has a wire speed of 40 Gbs and can deliver around 3.5 gigabytes per second over MPI. Since the system is too large to fit within a single switch chassis, multiple switches are used and they are connected together via 12x QDR links (120 Gbs) configured as a fully connected mesh topology. This topology is less expensive than a traditional full fat tree network yet still provides a relatively high bisection bandwidth.

The iDataplex system provides several innovative design features to improve energy efficiency. For example, the nodes are about half the depth of typical rack mount servers. The shorter depth results in more

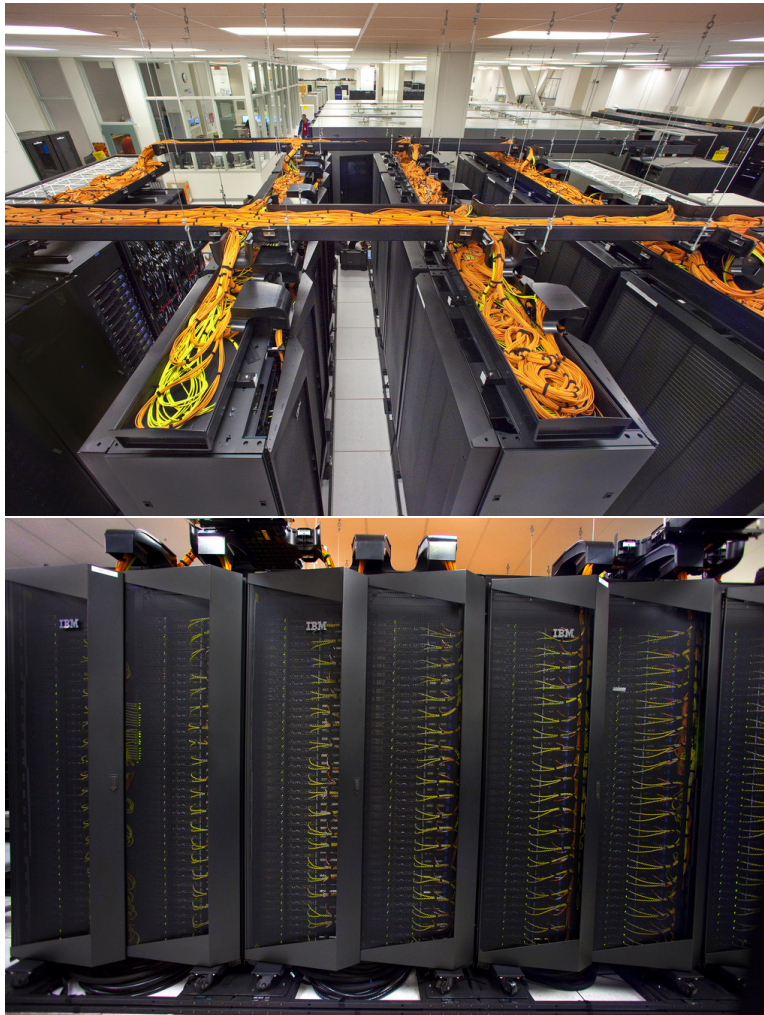


Figure 3.4: Photos of the Magellan System at NERSC. The top image shows the overhead cabling system which is suspended from the ceiling to seismically isolate the racks from each other. The lower image shows three racks of compute nodes. The seismic isolation platforms are visible below the racks.

efficient and uniform airflow since there is less air resistance. Additionally, two nodes are joined into a common node package. This allows larger, more efficient fans to be used. The system also allows for water-cooled doors, which are described below.

Cooling Design

Rear-door heat exchangers (RDHX) are installed on the iDataPlex system which improves energy efficiency. This cooling system removes the heat generated by each cabinet before the air is exhausted to the room, so the entire system is room neutral. For the configuration at NERSC the system is actually able to remove more heat than is generated by the rack, so it effectively assist in cooling the room. The RDHXs are connected to two cooling distribution units (CDUs). The CDUs distribute chilled water to each door and monitor the entrance and exit water temperatures to maintain temperatures above the dew point for the room. NERSC worked with the CDU manufacturer to modify the design of the CDU to directly use the building chilled water to feed the RDHXs versus the typical approach which uses a secondary chilled water loop. This modification improves the efficiency. Furthermore, the configuration at NERSC uses the return water from the Franklin Cray XT4 system. The Cray XT4 system must be operated at very low chilled water temperatures and the return water temperature is still relatively cool. Utilizing the return water improves the overall efficiency of the facility.

Storage Hardware

A high-performance scalable storage system was deployed as part of the testbed. The storage is configured to run IBM's high-performance parallel file system, GPFS. The storage hardware consists of four scalable storage units. Each storage unit consists of a DataDirect S2A9900 Controller Couplet with 300 1 TB SATA drives. Each unit is capable of delivering over 5 GB/s of peak bandwidth. The system is configured in a RAID6 configuration for maximum redundancy. Drives are organized in an 8+2 configuration. There is approximately 240 TB of RAID storage in each scalable unit. The total system provides almost 1 PB of total storage. The existing NERSC SAN was augmented to accommodate the additional storage. Nine Magellan I/O nodes are connected to the SAN and dedicated to providing GPFS services. The file system is mounted across the cluster and service nodes. These file systems are also shared across the center. The service nodes include the network nodes that will be used for testing of the Advanced Networking Initiative (ANI).

NERSC also expanded the archival storage system. A Sun (now Oracle) SL8500 library was installed and integrated with three existing SL8500 libraries. Each SL8500 is capable of holding 10,000 tapes. The library also included 35 T10K-C tape drives. The T10K-C can store 1 TB of uncompressed data on a single tape. Consequently, a single SL8500 library can potentially store up to 10 PB of uncompressed data. The hardware is managed by the High-Performance Scalable Storage (HPSS) software [7].

3.2.2 Software Stack

Flexibility was a major objective in selecting the components of the software stack. Since Magellan will be used to explore a variety of cloud service models and programming models, it was critical that the software stack enable the system to be quickly reconfigured. The cluster is managed using a combination of IBM's xCAT cluster management software and Adaptive Computing's Moab Adaptive Computing Suite. These two tools will allow us to provision nodes in the cluster from traditional batch environments to Amazon EC2-like models to Hadoop-based MapReduce clusters. NERSC is working with Adaptive Computing to explore how their software can be used to dynamically re-provision hardware between the different environments based on demand, energy consumption, utilization, as well as other factors.

IBM's xCAT cluster management software [12] provides a framework for provisioning and managing large clusters. xCAT can provision nodes using a variety of methods including diskless, disk-full, and hybrid. The framework can also utilize the IPMI management interface to automatically power on and off nodes. In addition to providing command-line tools, xCAT also provides an API that can be used by other programs such as Moab to remotely control the cluster.

The Adaptive Computing Suite from Adaptive Computing provides resource management and a highly capable scheduler to schedule and place workloads. The Moab Computing Suite represents Adaptive Computing's foray into the cloud computing arena. It aims to provide a suite of tools that can be used to manage private clouds. It provides an array of capabilities including integration with xCAT, green computing capabilities, and support for virtual private clusters. The integration with xCAT enables Moab to automatically provision nodes based on user job requirements. Moab can do bare metal provisioning with xCAT or use xCAT's hypervisor extensions to provision virtual machines. Moab can also interact with xCAT to power off nodes when they are idle. While NERSC traditionally runs with very high utilization, this capability is still interesting. Virtual Private Clusters (VPCs) offer another service model for offering compute resources. A VPC can include a suite of services and compute elements to create an entire cluster. For example, a VPC might include a head node running a private job scheduler, a database server, and a scalable number of compute nodes. Eventually, VPCs could be integrated with network services to create a virtual private network connection to other resources. This concept is described in more detail in Section 3.5.1.

Eucalyptus

The Eucalyptus project was first started to provide an open-source platform for conducting cloud computing research. It was intended to be API compatible with Amazon's EC2. A commercial company has since been started to continue the development of Eucalyptus and support enterprise customers using Eucalyptus to operate private clouds. Eucalyptus supports nearly all of the core features of EC2 including creating virtual instances, elastic block storage (EBS), S3, and elastic IP addresses. Since the software is open-source, it is possible to make modifications and add hooks and callouts to gather data. This makes it a useful tool for exploring cloud computing for the DOE SC.

In this reporting period, we deployed Eucalyptus on a 40-node test cluster and experimented with the software. Eucalyptus has a number of networking modes and we have experimented with the suitability of each of the modes. We have experimented with managing virtual clusters on top of Eucalyptus, tested the creation and deployment of new images, etc. Eucalyptus provides a convenient platform for creating and managing a private cloud platform for multiple users. However Eucalyptus, like much of cloud computing software, is still in its infancy and has design principles that may not be compatible with supercomputing center policies. In addition, bugs of various flavors have been encountered. The commercial company is currently focused on supporting its commercial offering and is limiting which capabilities it incorporates in the open version. The future roadmap of Eucalyptus and other cloud software is somewhat unclear at this time. ANL and NERSC plan to continue with Eucalyptus but are also evaluating alternatives such as Nimbus. ANL and NERSC are also reaching out to other cloud related projects outside of DOE such as NASA's Nebula project and the NSF FutureGrid to explore potential synergies.

Hadoop

Hadoop (described in 2.1) is attracting the attention of multiple scientific communities. From the user survey and discussions with users, Hadoop was cited as an area of high interest. A 40-node Hadoop cluster was deployed shortly after the Magellan hardware completed acceptance. This has provided an early testbed for users to explore the Hadoop model for their applications. Much of the early testing has come from the bioinformatics community, but other groups have also expressed an interest.

3.3 Benchmarking

Scientific applications can be broadly classified as a) *tightly coupled computations*, or b) *asynchronous or loosely coupled computations*. Tightly coupled applications are MPI codes that range from mid-range to large-scale. Mid-range applications that need tens to hundreds of processors often run in local clusters, while the large-scale codes run in supercomputing centers.

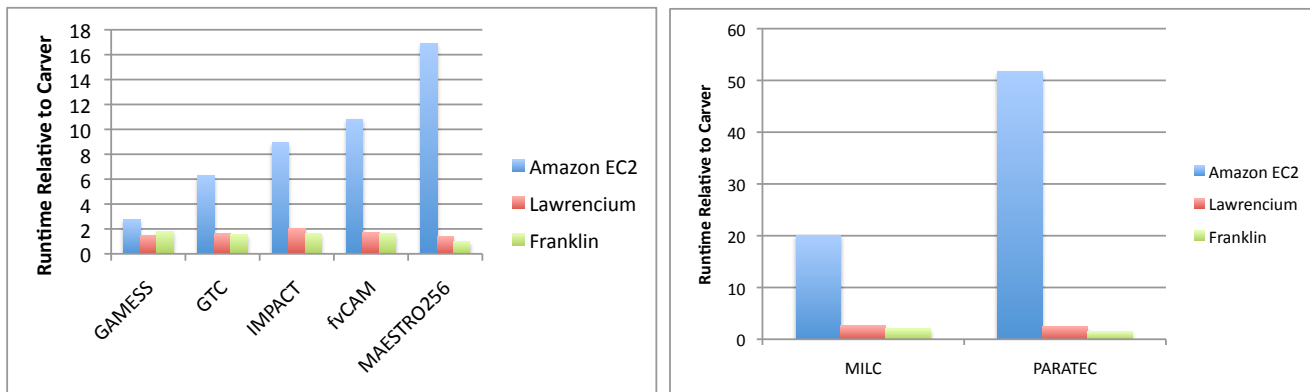


Figure 3.5: Runtime of each application on EC2, Lawrencium and Franklin relative to Carver.

Some scientific explorations are performed on the desktop or local clusters and have asynchronous or loosely coupled computations. However, the increased scale of digital data due to low-cost sensors and other technology [8] has resulted in the need for these applications to scale to environments such as cloud environments. The requirements of such applications are similar to those of the internet applications that currently dominate the cloud computing space, but with far greater data storage and throughput requirements.

It is important to understand the behavior of these classes of applications in cloud environments. We have performed a number of benchmarking experiments to understand the performance of applications in virtualized cloud environments and compared it to traditional HPC systems.

3.3.1 Experiment Setup

NERSC has a diverse user community with over 3,000 users, 400 distinct projects and around 600 codes from the DOE Office of Science research community. The NERSC6 benchmarks is a set of application benchmarks that span a range of science domains, parallel algorithms and concurrencies that has been selected to represent the user workload based on detailed workload characterization. For our benchmarking, we use a select set of the NERSC6 application benchmarks and High Performance Computing Challenge (HPCC) benchmarks.

The benchmarking was conducted in two phases. In Phase 1, we compared the performance of a commercial cloud platform (Amazon EC2) with HPC systems (Carver and Franklin) at NERSC and a mid-range computing system (Lawrencium) at LBNL. The Carver and Magellan systems are identical in hardware configuration and the traditional cluster configuration for Magellan uses the same software configuration as Carver. Here we report the results for Carver but they apply equally for Magellan when it is operating as a traditional cluster. The typical problem configurations for the NERSC6 benchmarks are defined for much larger systems. We constructed reduced problem size configurations to target the requirements of mid-range workloads.

In Phase 2, we are repeating the experiments of the reduced problem size configurations on Magellan hardware to further understand and characterize the virtualization impact. This is an ongoing effort and we report on early results.

3.3.2 Evaluation of Performance of Commercial Cloud Platform

Figure 3.5 shows the relative runtime of each of the test applications relative to Carver, which is the newest, and therefore fastest, machine in our testbed. For these applications, at these concurrencies, Franklin and Lawrencium are between $1.4\times$ and $2.6\times$ slower than Carver. For EC2 the range of performance observed is significantly greater. In the best case, GAMESS, EC2 is only $2.7\times$ slower than Carver. For the worst case,

Magellan Progress Report

Table 3.3: HPCC Performance

Machine	DGEMM Gflops	STREAM GB/s	Latency μ s	Bandwidth GB/s	RandRing Lat. μ s	RandRing BW GB/s	HPL Tflops	FFTE Gflops	PTRANS GB/s	RandAccess GUP/s
Franklin	8.4	2.30	7.8	1.6	19.6	0.19	0.47	14.24	2.63	0.061
Lawrencium	9.6	0.70	4.1	1.2	153.3	0.12	0.46	9.12	1.34	0.013
EC2	4.6	1.7	145	0.06	2065.2	0.01	0.07	1.09	0.29	0.004
Magellan VM	7	2.9	78.7	0.47	211.82	0.01	0.25	1.82	0.46	0.009
Magellan	10.2	3.8	1.81	3.48	5.17	0.31	0.57	29.58	9.56	0.040

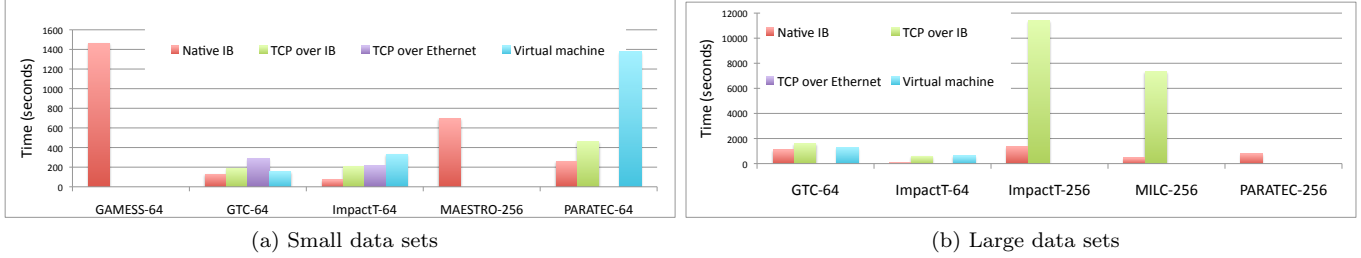


Figure 3.6: Preliminary results from experiments running NERSC applications on the Magellan testbed. Data is still being collected for some of the data points at the time of this writing. Also the data collected on the virtual machines were collected on an isolated testbed with less contention.

PARATEC, EC2 is more than $50\times$ slower than Carver. This large spread of performance simply reflects the different demands each application places upon the network.

These experiments and results are detailed in a manuscript in submission that is included in the Appendix.

3.3.3 Comparison of Virtualization and Bare Metal Performance on Magellan

We are conducting experiments on Magellan hardware to understand the impact of virtualization and other cloud features on scientific applications.

The Magellan testbed enables us to exercise control on different configurations including compiler versions and communication protocols. This allows us to study the impact of different parameters in greater detail.

Figure 3.6a shows the execution time on the smaller data sets set up for cloud evaluation in the previous study. The graphs show preliminary data collected on the Magellan testbed and experiments are still being conducted. We compared the application execution time on batch queue systems a) where the application used native IB, b) where the application used TCP over IB for the communication, c) where the application used TCP over Ethernet for communication, and d) where the application was run on a virtual machine. The virtual machine data was collected on an isolated testbed whereas the other data was collected on the shared production system. We will repeat these experiments on the isolated testbed to better understand the impact of various factors. Figure 3.6b shows the execution time for larger sized NERSC6 benchmark runs. As expected, the communication sensitive applications perform better with TCP over IB than TCP over Ethernet due to the higher bandwidth. The performance of the applications on the virtual machines is affected significantly for applications such as PARATEC that are communication sensitive and IMPACT-T at a larger scale of 256 cores.

3.3.4 HPCC

The results of running HPCC v.1.4.0 on 64 cores on the machines in our study are shown in Table 3.3. The DGEMM results are consistent with the processor performance as one for the benchmark. The STREAM results show that EC2 is significantly faster for this benchmark than Lawrencium. We believe this is because of the particular processor distribution we received for our EC2 nodes for this test. We had 26 AMD Opteron

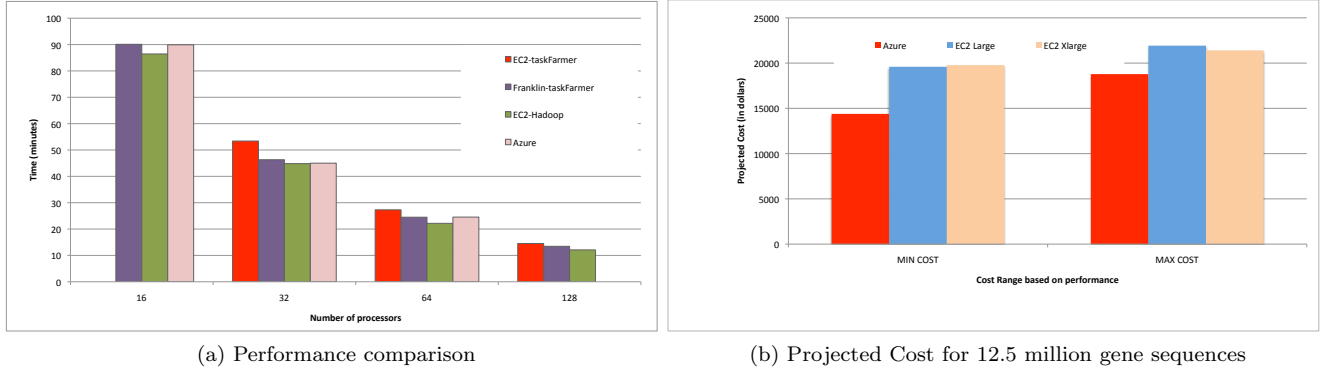


Figure 3.7: Performance and cost analysis of running BLAST on a number of cloud platforms

270's, 16 AMD Opteron 2218 HE's, and 14 Intel Xeon E5430's, of which this measurement represents an average. The AMD Opteron based systems are known to have better memory performance than the Intel Harpertown processors used in Lawrencium. Both EC2 and Lawrencium are significantly slower than the Nehalem-based system, however.

The network latency and bandwidth results clearly show the difference between the interconnects on the tested systems. The uncontended latency and bandwidth measurements of the EC2 Gigabit Ethernet interconnect are more than 20 times worse than the next slowest machine. Both EC2 and Lawrencium suffer a significant performance degradation when self-contention is introduced, as demonstrated by the Random Ring results. The EC2 latency is 13 times worse than Lawrencium, and more than 400 times slower than a modern system like Magellan. The bandwidth numbers show similar trends: EC2 is 12 times slower than Lawrencium, and 30 times slower than Magellan.

HPL is the high-performance version of the widely reported Linpack benchmark, which is used to determine the TOP500 list. It solves a dense linear system of equations and its performance depends upon DGEMM and the network bandwidth and latency. On a typical high performance computing system today, roughly 90% of the time is spent in DGEMM, and the results for the three HPC systems illustrate this clearly. However, for EC2 the less capable network clearly inhibits overall HPL performance, by a factor of six or more. The FFTE benchmark measures the floating point rate of execution of a double precision complex one-dimensional discrete Fourier transform, and the PTRANS benchmark measures the time to transpose a large matrix. Both of these benchmarks' performance depends upon the memory and network bandwidth and therefore show similar trends. EC2 is approximately 20 times slower than Carver and four times slower than Lawrencium in both cases. The RandomAccess benchmark measures the rate of random updates of memory and its performance depends on memory and network latency. In this case EC2 is approximately 10 times slower than Carver and three times slower than Lawrencium.

Table 3.3 also shows the performance of the HPCC benchmarks on the Magellan hardware. DGEMM and the synthetic benchmarks (HPL, FFTE and RadomAccess) show better performance on our VM compared to EC2. The Ping Pong Latency and Bandwidth is better on our VM, but the RandomRing latency and bandwidth is similar to EC2.

Overall the results of the HPCC runs indicate that the lower performing network interconnect in virtual machines has a significant impact upon the performance of even very simple application proxies. This is illustrated clearly by the HPL results which are significantly worse than would be expected from simply looking at the DGEMM performance.

3.3.5 Comparison of Performance of BLAST

The Joint Genome Institute’s Integrated Microbial Genomics (IMG) pipeline is a motivating example of a workload with growing needs for computing cycles due to the growth of sequence data expected from next-generation sequencers. We benchmarked the BLAST algorithm on HPC systems (Franklin at NERSC) and cloud platforms including Amazon EC2, Yahoo M45! and Windows Azure. We managed the execution of loosely coupled BLAST runs using a custom developed task farmer and Hadoop.

For this evaluation, we compared the performance data of running 2500 sequences against a reference database of about 3 GB. Figure 3.7 shows the performance of the task farmer on Franklin and Amazon EC2 and the performance of Hadoop on EC2 and the BLAST service on Windows Azure. For this workload, the performance of cloud platforms compares with traditional HPC platforms.

On Amazon EC2, the database and the input data sets were located on elastic block store (a high-speed efficient and reliable data store provided by Amazon for use with one node) that is served out to the other worker nodes through NFS.

We were provided friendly access to the Windows based BLAST service deployed on Microsoft’s cloud solution, Windows Azure. The experiments were run on Azure’s large instance that has four cores/1.6 GHz and 7 GB of memory. This instance type is between the large and xLarge instances we used on Amazon EC2. On Azure, the total time for data loading into the virtual machines from the storage blob and back to the storage blob is also included in the timing. Additionally the data sets were partitioned equally across the nodes and we see the effects of bunching. Thus the model here is slightly different from our setup on Hadoop but gives us an idea of the relative performance.

We ran the same experiment on Yahoo! M45, a research Hadoop cluster. The performance is comparable to other platforms with a small number of sequences. The load on the system affects execution time, inducing large amount of variability, but performance seems to be reasonable on the whole. However, when running 250 concurrent maps (10 genes per map, 2500 genes total), there is a heavy drop in performance – the overall search completes in 3 hours and 39 minutes. This seems to be the result of thrashing since two map tasks, each requiring about 3.5 GB, are run per node with 6 GB memory.

We extrapolated from the running times the cost of commercial platforms to run a single experiment of 12.5 million sequences (run every 3 to 4 months). The cost of a single experiment varies from about \$14K to \$22K dollars based on the type of machine and the performance achieved on these platforms (Figure 3.7).

There is a reasonable amount of effort required to learn these environments and to get these environments set up for a particular application. Early results of this work were presented at the System Biology Cloud Workshop at SC09. A manuscript is in preparation with additional experiments that explores the different data storage and communication options.

3.3.6 Hadoop Benchmarks on Magellan

We benchmarked the Hadoop Magellan cluster with available Hadoop tests. The benchmarking data enables us to set a baseline to compare against when performing optimizations for scientific workloads in the future.

MRBench evaluates the performance of MapReduce systems while varying key parameters such as data size and the number of Map/Reduce tasks. We varied the number of lines of data written from 100 to 1000 and varied the number of maps and reduces. Figure 3.8 shows the time with varying maps and reduces for a) 100 and b) 1000 lines. As the number of maps and reduces increases the time taken increases; however the difference is less than 10 seconds. The number of lines written at the orders of magnitude we measure shows no perceptible effect. MRBench can be provided with custom mapper and reducer implementations to measure specific system or application behavior. There is a need to develop benchmarks that emphasize the nature of scientific workloads.

TestDFSIO measures the I/O performance of HDFS. Figure 3.9 shows the throughput for small and large file sizes with varying concurrent writers/files. For small file sizes, the throughput remains fairly constant with varying number of concurrent writers. However, the throughput decreases rapidly as the number of concurrent files/writers increases. This seems to be dependant on the HDFS block size and will require tuning and additional benchmarking to understand the configuration necessary for scientific applications.

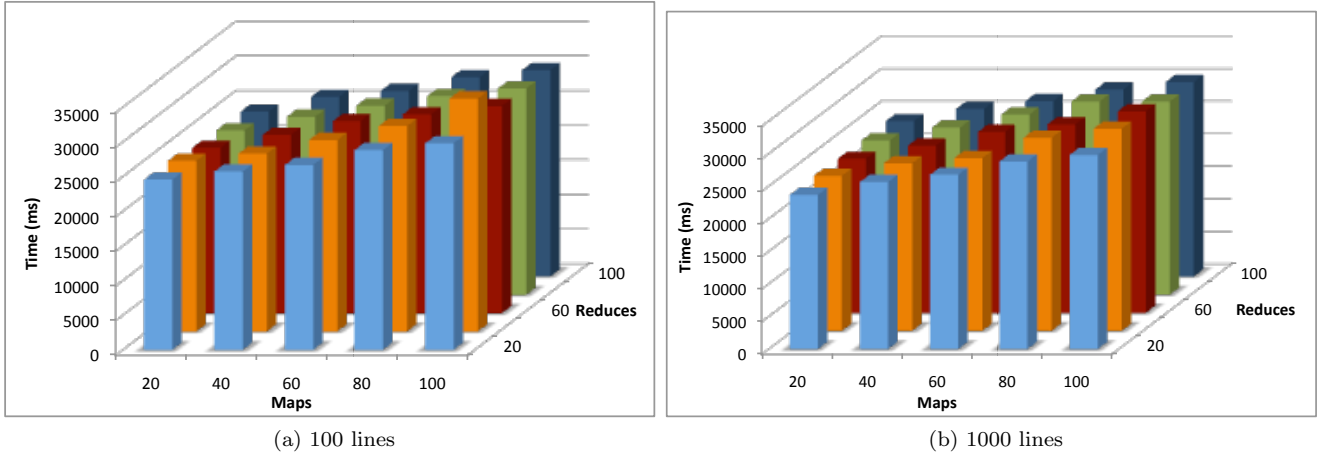


Figure 3.8: Hadoop MRBench.

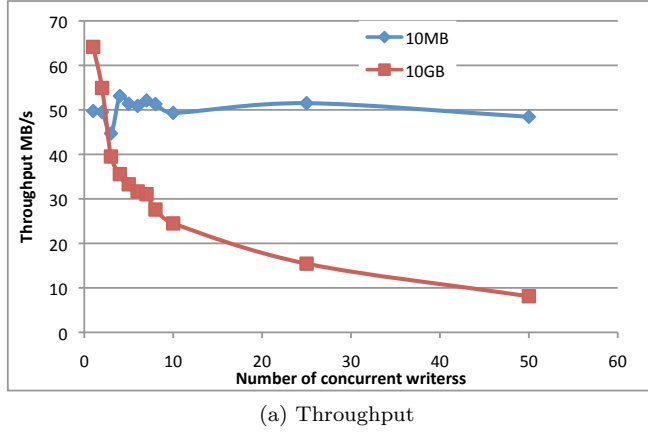


Figure 3.9: HDFS throughput for different file sizes with varying number of concurrent writers.

Table 3.4: Terasort

Program	Time (minutes)
TeraGen	42
TeraSort	91

TeraSort is a standard map/reduce application for Hadoop that was used in the terabyte sort competition. TeraGen generates the data, and TeraSort then samples the input data and uses map/reduce to sort the data in total order. Table 3.4 shows the timing for running TeraGen and TeraSort on our 40-node (312 cpu) cluster with 64 reduce tasks.

The Hadoop benchmark enables us to benchmark some base system parameters. However, the nature of scientific workloads using Hadoop is largely unknown today. Thus further understanding and tuning of system parameters will be needed to manage Hadoop workloads.

3.3.7 Summary of Benchmarking

Applications that perform the most collective communication and send the most messages perform the worst on EC2 and Magellan virtual machines. We plan to run additional experiments at larger scales and instrument applications with IPM to understand the impact on mid- to large-sized scientific codes on virtualized and non-virtualized cloud environments. IPM provides a low-overhead performance profile of the performance aspects and resource utilization of scientific applications. In addition, we are planning additional benchmarking with user codes. A special queue on Magellan will enable users to collect benchmark data with IPM turned on. Similarly users will be provided access to Eucalyptus to compare the performance of their applications on virtual environments.

3.4 Flash Storage Evaluation

Solid-state storage (SSS) is poised as a disruptive technology. This impact would likely affect both the cloud computing and scientific computing spaces. For these reasons, flash storage evaluation was included in the Magellan project. Magellan at NERSC is first evaluating several products and will deploy a larger storage system based on this evaluation. The technologies being evaluated include three PCI-e connected solutions and two SATA connected solutions. Table 3.5 summarizes the products that are under evaluation.

NAND flash devices have dramatically different performance characteristics compared with traditional disk systems. NAND typically delivers much higher random read rates. However, NAND chips must be erased prior to writing new data. This erase cycle can be extremely slow. For example, NAND may require several milliseconds to erase a block, yet can perform a read of block in several microseconds [11]. To help mask this impact, many high-end devices utilize background grooming cycles to maintain a pool of erased blocks available for new writes.

While one expects the superior performance of flash, particularly for random read operations, to lead to improved application performance, experience shows that this is not always the case. Often code paths in the kernel or applications have not been sufficiently optimized to take advantage of the performance the new devices can deliver. In some cases, this is due to decades worth of optimization to deal with the distinct performance characteristics of disk. Swap algorithms and IO schedulers will often buffer and flush data to minimize the number of writes. However, these optimizations can increase the latency and prevent the flash devices from maintaining a sustained stream of IO operations. Additionally, many of the devices benefit from a high number of threads issuing writes. This prevents the buffer queues from being exhausted too quickly. For disk systems, this can have an adverse effect of creating extra head movement, which slows down throughput. The difference in the performance characteristics between disk and solid state storage make it clear that applications need to be evaluated to see what near-term benefit SSS can provide. Eventually, improvements in the algorithms in the kernel, middleware, and applications may be required to realize the full potential.

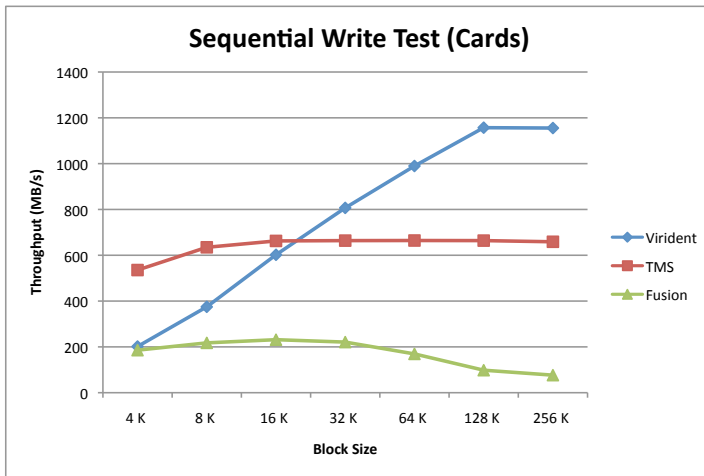
Benchmarking efforts to date have focused on measuring the bandwidth the devices can deliver. Plots in Figure 3.10 show the performance of the devices across a range of blocks sizes for both sequential read and write. Ongoing benchmarking is focused on identify how the devices behave under a sustained heavy random-write load to address the impact of any grooming cycles. Of the card-based solutions, this impact is most prevalent on the FusionIO device, where the bandwidth can drop over an order of magnitude during grooming. For applications with very high-burst IO followed by several minutes of idle IO, this grooming cycle could be hidden from the application. However, few applications are likely to be so well behaved, and mixed workloads further complicate the picture.

3.5 Usage and Impact

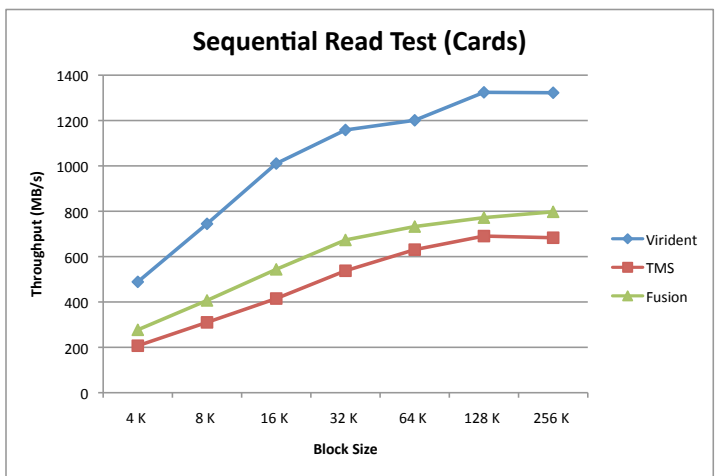
The Magellan system is already having an impact. The Magellan hardware was immediately placed into service running as a traditional computing cluster. This allowed NERSC users to quickly take advantage

Table 3.5: Summary of flash-based storage products that were evaluated.

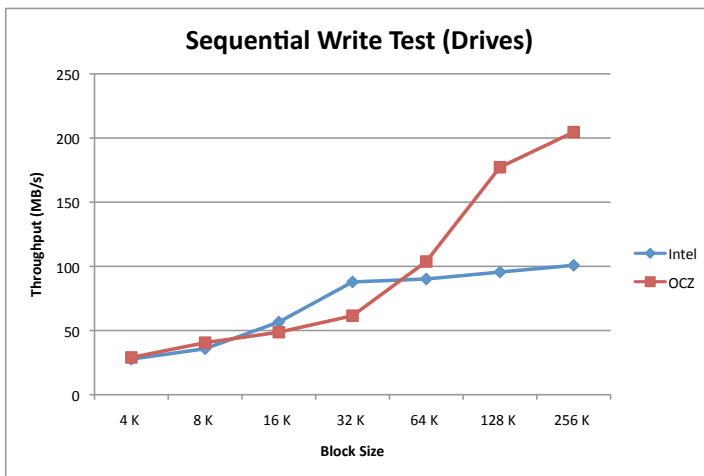
Manufacturer	Product	Capacity
PCI-e Attached Devices - All use SLC Flash		
Texas Memory Systems	RamSAN 20	450 GB
FusionIO	ioDrive Duo	320 GB
Virident	tachION	400 GB
SATA Attached Devices - Both use MLC Flash		
Intel	X25-M	160 GB
OCZ	Colossus	250 GB



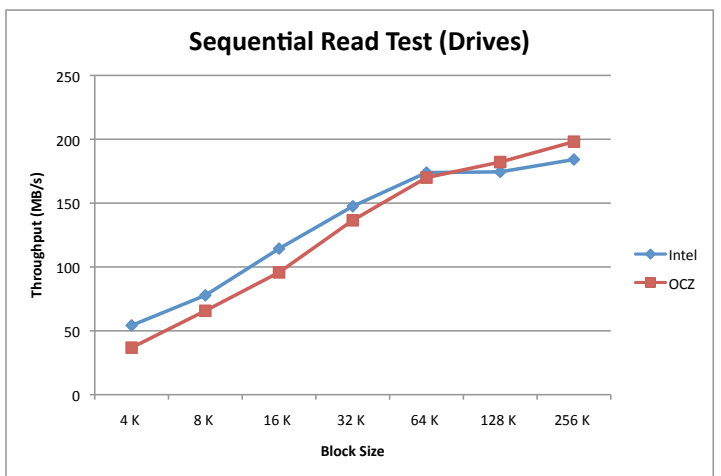
(a) Write Test for PCI-e attached Flash Storage



(b) Read Test for PCI-e attached Flash Storage



(c) Write Test for SSDs



(d) Read Test for SSDs

Figure 3.10: Performance of Sequential Read and Write IO Benchmarks performed on PCI-e attached flash storage devices and SSDs. The plots are for 16 concurrent threads issuing IO with varying block size.

of the system and prevent the system from being under-utilized while new cloud models are configured and deployed on the system.

3.5.1 Hardware as a Service – Joint Genome Institute

Hardware as a Service (HaaS) offers one potential model for supporting the DOE’s scientific computing needs. This model was explored early with Magellan when a facility issue at the Joint Genome Institute (JGI) led to their need for rapid access to additional resources. NERSC responded to this by allocating 120 nodes of Magellan to JGI.

Using ESnet’s OSCARS [6] and the Science Data Network, nine 1 Gb Layer 2 circuits were provisioned between NERSC’s Oakland Scientific Facility (OSF) and the JGI Walnut Creek Facility. In essence, the JGI internal network was extended twenty miles away to the Oakland facility. These two locations are adjacent on the ESnet Bay Area MAN (Metropolitan Area Network), and therefore there was ample bandwidth with relatively low latency. Dedicated switches at OSF were connected to the layer 2 connection, and the allocated compute nodes were connected to this network. To maintain separation, the InfiniBand network was disconnected on the allocated nodes. OS provisioning was handled using JGI’s existing management infrastructure, and the allocated nodes were booted over the Layer 2 network. As a result, the nodes had immediate access to all the critical data, databases, and account management services at JGI. This included accessing NFS file servers located in Walnut Creek. Within days, JGI users were running jobs on the allocated hardware. The hardware and network proved extremely stable. No changes were required on the part of the users. They simply submitted to the existing Sun GridEngine scheduler as usual. Jobs were transparently routed to the nodes in Magellan. This effort not only demonstrated the validity of hardware as a service, it enabled JGI to continue operations through a potential crisis with no impact on production sequencing operations.

While this demonstration was a success, the experience illustrated several areas for improvement. Ultimately, true hardware as a service requires a nearly automated, on-demand ability to provision hardware and create the necessary network connections to remote resources. For example, the requirement to disconnect the InfiniBand network should be handled by software. This could potentially be addressed by disabling ports on the switch or in the subnet manager. Alternatively, virtualization could be used to create virtual machines that would aid in the separation of resources. Another improvement would be to automate the provisioning and configuration of the network. ESnet’s OSCARS provides much of this capability, but it would need to be integrated with the resources manager like Moab and GridEngine. Eventually, one could envision a model where a system administrator at a DOE site would identify the need for more resources. With a few simple commands, the administrator could request resources from Magellan. The Magellan scheduler would allocate the resources and communicate with OSCARS to provision the network link between the two distant sites. Finally, Magellan would provision the nodes, either using a user-provided image or by directly booting from management services at the home site. NERSC and ESnet are working together to explore this model with the goal of demonstrating the capability using Magellan.

3.5.2 Traditional Cluster Usage

The Magellan system has been available for use in a traditional HPC cluster configuration since acceptance was completed. Users have accumulated over 12.8 million core hours between March 17 through June 24, 2010. Users from all offices have made use of the system in that time (Table 3.6 and Figure 3.11a). However, BES accounted for 70% of the utilization, followed by BER with 13%. The high utilization by BES is primarily due to several material science and chemistry projects taking advantage of the system. Many of these projects use mid-range applications that are particularly well suited to a system like Magellan. The top two users were both from BES and accounted for 30% of the overall usage. A plot of the usage at different job size is shown in Figure 3.11b. The plot illustrates that much of the usage is consumed by jobs running at 128 cores or larger (79%).

Table 3.6: Summary of usage of Magellan as a traditional cluster by office from March 17, 2010 through June 25, 2010.

Office	Raw Hours	Percent
Basic Energy Sciences	8,852,300	69%
Biological & Environmental Research	1,683,150	13%
High Energy Physics	1,169,420	9%
Fusion Energy Sciences	582,004	5%
Advanced Scientific Computing Research	403,169	3%
Nuclear Physics	104,362	1%

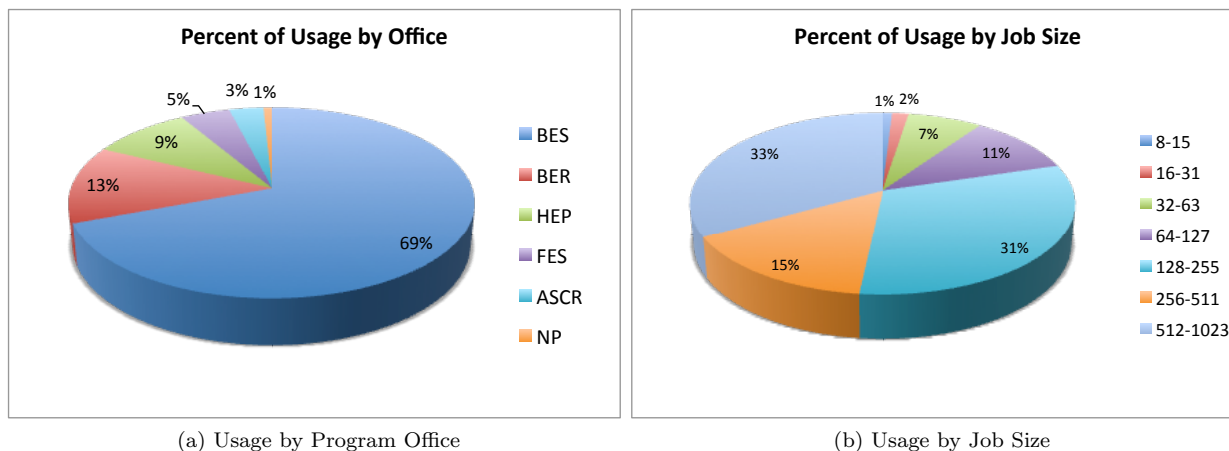


Figure 3.11: Usage by DOE Program Office and job size on Magellan as a traditional cluster between March 17, 2010 and June 24, 2010.

Chapter 4

Challenges and Future Plans

4.1 Challenges

4.1.1 Security and User-Provided Images

The majority of cloud related security analysis and security research has been from the perspective of the cloud service consumer. However, the problems and concerns are quite different for a cloud provider. The ability for users of the system to have local root privileges on the booted virtual machine instances fundamentally changes how those systems should be treated and secured. In many ways, the challenges of securing these virtual machines overlap with the existing challenges of securing a general use batch execution environment. However, when users are able to bring in their own system images, many of these problems are exacerbated. Furthermore, new methods for securing the systems are needed. Challenges created by this can be broadly placed into two major groups: system access and network communications.

System access can be broadly thought of as controlling who is able to access a computer system and what services are provided. One of the most fundamental controls typically provided by system ownership is the ability to decide who is allowed to access that resource. This is true in terms of user accounts as well as services which might provide privileged and unprivileged access to the system. Account creation represents a very real threat in a traditional notion since controlling user accounts is one of the most basic ways of controlling access. Once users are allowed to bring in their own system images, it is necessary to give up absolute control of user accounts. System integrity is another aspect which represents a real challenge. In essence, the responsibility and risk of securing and maintaining an image shifts from resource administrators to the image providers. If an image is poorly maintained or configured and provides remote access (intentionally or unintentionally), it may be possible to gain privileged access to the system via local vulnerabilities. Additionally, there may be issues of resource stability if untested software is put together on the part of the user.

Network communications describes the set of issues related to providing remote access (such as web services) as well as limiting where a system is able to connect to. As alluded to previously, a system image may include un-patched or misconfigured applications which can create a security threat when the image is instantiated. For any but the most transient services, such applications are rapidly discovered by scanning and exploited by automated bots. Additionally, with root access comes the ability to control how a system communicates in terms of system routing, tunneling and access to raw sockets. The socket problem is particularly troublesome in that it provides the ability to attack the actual Ethernet infrastructure in a way which is outside the traditional threat model.

Fortunately, addressing these challenges is not as difficult as it might otherwise seem. A combination of both of policies and software controls can be employed. It is possible to perform basic checks against system images that will test for known problems and to incorporate these checks into the image registration process. In addition, users may be required to run specific versions of certain software. For example, NERSC

uses a modified version of the SSH Daemon which captures keystroke data and looks for suspicious behavior typically observed in compromised systems. Also, the resource provider controls network and system access to the entire cloud infrastructure. This presents opportunities for both monitoring and limiting network or system access. Finally, educating the user base on appropriate activity as well as providing simple tools for testing and validation may be the most effective action we can take. Ultimately, both the resource providers and the scientific users share a common desire to maximize the use of the resources for scientific discovery.

4.1.2 InfiniBand Support for Virtual Machines

The performance evaluation in Section 3.3 illustrates the importance of a high-bandwidth, low-latency network. However, combining virtual machines and InfiniBand network presents several challenges. Possible methods for extending the InfiniBand network inside a virtual machine include: routing IP over IB from the Hypervisor to the virtual machine; bridging Ethernet over IB through the Linux bridge to the virtual machine; and full virtualization of the NIC. We will briefly discuss these options, including the benefits and tradeoffs and the current issues.

Routing IP packets from the virtual machine (VM) to the hypervisor (HV) and then using IP over IB is the most readily available approach today. In this approach, the VM communicates through the HV using a virtual network interface. Then standard IP routing is used to route the packets over the IB interface. This approach provides the lowest performance of the various options. Since the packet must traverse multiple TCP stacks (in the VM and through the HV), the overhead is high. There are also some challenges in integrating this with cloud management software like Eucalyptus.

Utilizing Ethernet over IB is another approach. In EoIB, first proposed by Mellanox, the Ethernet frames are encapsulated in an InfiniBand packet. The InfiniBand stack presents a virtual Ethernet NIC. This virtual NIC would be presented on the HV and then bridged to the VM. This approach has several advantages over the other approaches. VLANs should be supported in this model, which would provide some support for isolating networks. Bridging of the virtual Ethernet interface is similar to how Ethernet NICs are commonly bridged for a virtual machine. This means that packages like Eucalyptus could more easily support it. Since the communication would be TCP based, some performance would be lost compared with remote memory access offered by native InfiniBand. However, the packets would only go through a single TCP stack (on the VM) on each side. So the performance should exceed the IP over IB method. While this approach looks promising, the current EoIB implementation doesn't support Linux bridging. NERSC is in discussions with Mellanox about adding support for this. Another disadvantage is that only Mellanox currently plans to support EoIB, and it requires a special Bridge device be connected to the InfiniBand network. However, the devices are relatively low cost.

Full virtualization of the IB adapter is just starting to emerge. This method would provide the best performance, since it would support full RDMA. However, it would be difficult to isolate networks and protect the network from actions by the VM. Furthermore, images would have to include and configure the InfiniBand stack. This would dramatically complicate the process of creating and maintaining images. Finally, only the latest adapters from Mellanox are expected to support full virtualization. The version of adapters in Magellan will likely not support this method. We are exploring options for evaluating virtualization of the NIC and looking at alternative methods.

High-performance networks like InfiniBand can dramatically impact the performance, but integrating them into virtual machine based systems is difficult. It is unclear at this point if improving this support is the best approach. Alternatively, it may be more attractive to look at how some of the benefits of virtualization can be provided to the users without using true virtualization. For example, can dynamic imaging of nodes or software overlay methods be used to allow the user to tailor the environment for their application? This is one of the key questions that Magellan will continue to explore.

4.1.3 Secure Access to Shared File Systems from Virtual Machines

HPC systems typically provide a high-performance scratch file system that is used for reading initial conditions, saving simulation output, or writing checkpoint files. This capability is typically provided by parallel file systems such as GPFS, Lustre or Panasas. Providing this capability for virtual machine based systems presents several challenges including security, scalability, and stability.

The security challenges focus around the level of control a user would have within the virtual machine. Most parallel file systems use a host based trust model, i.e., a server trusts that a client will enforce access rules and permissions. Since the user would likely have root privileges on the VM, they would be capable of viewing and modifying other users' data on the file system.

Accessing shared file systems from user controlled virtual machines can also create scalability and stability challenges. Since each node could potentially run many virtual machine instances, the file system could see a multiplier of file system clients. If each node was running eight small instances (one on each core) then the file system would have to deal with eight times more clients. Furthermore, virtual machines are often terminated versus performing a clean shutdown. This could lead to clients frequently joining and leaving the file system cluster. File systems like GPFS and Lustre employ timeout and heart beat methods that assume a relatively stable client pool. Clients randomly disappearing could lead to long hangs while outstanding locks held by terminated instances were timed out.

There are several potential methods to address these issues. One would be to project only the portions of the file system that the user owns into the virtual machine. This could be done using protocols like NFS and re-exporting the parallel file system. Another approach would be to forward the file IO operations to a proxy server that would have the file systems mounted. The operations would then be performed on the proxy server as the user who owned the virtual machine instance. The standard file system client would enforce the access controls.

There are existing file system modules that use the Linux FUSE file system interface to forward IO over connections like SSH. The performance over SSH would be poor, but would at least provide access to data. Alternative, the IOFSL project [1] is developing a high-performance IO forwarding layer that could potentially help. While IOFSL is focused on developing an IO forwarding system to support ultra-scale HPC systems, the same mechanisms can be used for virtual machines. These solutions would also help mitigate the scaling and stability challenges too since they would reduce the number of real clients handled by the file system and would act as a firewall between the virtual machines and the file system.

4.1.4 Hadoop Programming Interface

Hadoop applications are designed to use HDFS to get the benefit of data-location scheduling. This model can be used for scientific applications, but requires rewriting the applications to use the HDFS file system interface. Some projects have implemented native Hadoop based applications that take advantage of the full capabilities of the MapReduce model by using HDFS. For example, the authors of CloudBurst have implemented short read mapping for genomic sequence data in Hadoop [10]. However, rewriting applications to use a new file system interface is unlikely to be practical for most application groups, especially those with large legacy code bases that have undergone decades of validation and verification.

The Hadoop streaming model allows one to create map and reduce jobs with any executable or script as the mapper and/or the reducer. This is the most suitable model for scientific applications that have years of code in place capturing complex scientific processes. This approach does not benefit from the data locality aspects of Hadoop, but a number of groups are interested in the tool as a means to managing loosely coupled asynchronous runs. In addition, tools above Hadoop such as HBase are useful for data management and querying that has largely been handled through ad-hoc solutions in the scientific community.

4.1.5 Eucalyptus Accounting and Scheduling

In this reporting period, we deployed and tested Eucalyptus on our testbed. A number of our users have tested and experimented with Amazon's Web Services. Thus the interface compatibility of Eucalyptus with

Amazon makes it an attractive choice for us to enable direct extensibility of resources available to the end user both at NERSC and through private cloud providers.

As stated earlier, Eucalyptus is a convenient platform for creating and managing virtual machines. However, Eucalyptus does have some limitations. Eucalyptus' error checking and handling of failures can result in an inconsistent state of the system, often requiring a clean restart. This is clearly not feasible in a production environment that serves a large community of users.

The resource allocation and accounting model in Eucalyptus is different from the traditional supercomputing model. The resource control in the cloud model is explicit, where resources are allocated on demand at the instant the resources are requested. Batch queue systems such as Moab, PBS, etc. provide customizable policy points and algorithms to control scheduling of requests and resources. Similarly, there is a need for policy points in the cloud infrastructure and the ability to store and service requests when resources cannot be allocated immediately.

4.2 Future Plans

For the remainder of the Magellan Project, NERSC will focus on addressing many of the challenges discussed earlier and resolving the outstanding questions from the Research Agenda. Here is a brief list of the capabilities we plan to deploy and research questions we will study.

- *Continue to study how applications behave in cloud environments and what applications are best suited to running in current and emerging commercial cloud offerings.* Our early study indicates that applications with synchronous communication needs perform poorly in virtual cloud environments. We are conducting further benchmarking and testing to understand the impact of virtualization on different resource needs including memory, I/O, bandwidth, power. This will enable us to determine the efficiency of cloud computing with respect to DOE Office of Science workloads.
- *Utilize monitoring tools to collect data on application characteristics for jobs run on the traditional cluster configuration of Magellan.* We plan to use IPM to understand the characteristics of the application codes run at NERSC to identify the codes that might benefit from a cloud environment.
- *Deploy a limited number of specialized project images.* We will select a small number of projects and prepare special images that will run in on “bare metal” without virtualization. We will use this to evaluate how to automate provisioning on demand.
- *Provide capabilities for users to instantiate their own images while closely considering the security implications.* We will build and test a simple accounting model in Eucalyptus to track user usage and introduce simple scheduling policies to ensure fairness amongst the users. In addition we will apply security strategies and policies as discussed earlier.
- *Provide support for virtual private clusters.* Virtual private clusters are sought by a number of user groups at NERSC that would like dynamic access to resources for a specified period of time. Cloud technologies enable us to implement dynamic on-demand private clusters that allow users complete control on the resource set for a specific period of time.
- *Users will be given access to the flash resources.* We will evaluate models where the flash is accessed both as a requestable resource and as back-end storage device for network file system.
- *Evaluating networking options in cloud environments including how to best utilize InfiniBand in virtual machines.* Current offerings of cloud computing, both private and commercial, rely on Ethernet networks. We are exploring the networking options within virtual machines with the goal of improving the environment for scientific workloads.

- *Access to shared file systems.* The user survey indicated that users were interested in a cloud setup at NERSC that would let them leverage both the benefits of cloud environments as well as traditional supercomputing services such as access to parallel file systems and mass storage systems. We will begin exploring ways to securely provide access to shared file systems from virtual machines.
- *Enable applications to use Hadoop technologies.* Our early work shows that Hadoop and related technologies such as HDFS and HBase are useful for data-intensive sciences. We will work closely with the IMG pipeline and Supernova Factory to develop cost-effective porting of their applications to run in Hadoop.
- *Enable single front-end to multiple clouds.* We are working with application groups such as IMG and Open Science Grid (OSG) to access cross-site cloud sites. We will explore ways for applications to leverage cloud sites at Argonne, NERSC and commercial providers simultaneously.

Chapter 5

Conclusions

The mission of the Magellan Project is to understand the role of cloud computing in the resource needs for the Department of Energy's Office of Science (DOE SC). NERSC has adopted a data-driven approach and has deployed a flexible testbed that will be used to explore and evaluate various cloud computing technologies and solutions. The 720-node cluster is available to users through traditional batch queues and is already having an impact. NERSC has solicited input from users to understand their motivations and interest in cloud computing that has motivated our testbed software stack and design decisions. Existing NERSC users across all DOE Program Offices are already making heavy use of the system. In addition, we demonstrated successful deployment of Hardware as a service at the Joint Genome Institute. Hadoop and Eucalyptus have also been deployed on the Magellan system that is being used for benchmarking and testing as well as to support early cloud users. We have benchmarked the NERSC6 Application Suite and the HPCC Benchmark on a variety of architectures including traditional HPC systems, public and private cloud systems. The results to date illustrate the potential impact cloud computing can provide for DOE SC, but also point to some of the weaknesses and gaps in the current offerings. During the remainder of the project will we study these issues in more detail and explore potential approaches to addressing these challenges.

Acknowledgments

This work was funded in part by the Department of Energy from The American Recovery and Reinvestment Act of 2009 under contract number DE-AC02-05CH11231. In addition, parts of this work were funded by Office of Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-C02-05CH11231. The authors would like to thank Yahoo! and CITRIS, UC Berkeley for access to the Yahoo! M45 cluster. CITRIS, UC and Greg Bell provided Amazon EC2 access. The authors would like to thank Nick Wright, Shreyas Cholia, John Shalf, Harvey Wasserman and Krishna Muriki at NERSC and Wei Lu and Dennis Gannon at Microsoft for access to the prototype Windows Azure BLAST service.

Bibliography

- [1] N. Ali, P. Carns, K. Iskra, D. Kimpe, S. Lang, R. Latham, R. Ross, L. Ward, and P. Sadayappan. Scalable I/O forwarding framework for high-performance computing systems. In *IEEE International Conference on Cluster Computing (Cluster 2009)*, 2009.
- [2] S. Canon, S. Cholia, J. Shalf, K. Jackson, L. Ramakrishnan, and V. Markowitz. A performance comparison of massively parallel sequence matching computations on cloud computing platforms and hpc clusters using hadoop. In *Using Clouds for Parallel Computations in Systems Biology Workshop, Held at SC09*, 2009.
- [3] Cern Virtual Machines. <http://rbuilder.cern.ch/project/cernvm/releases>.
- [4] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. BigTable: a distributed storage system for structured data. In *OSDI '06: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*, pages 15–15, Berkeley, CA, USA, 2006. USENIX Association.
- [5] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI '04*, pages 137–150, 2004.
- [6] C. P. Guok, D. W. Robertson, E. Chaniotakis, M. R. Thompson, W. Johnston, and B. Tierney. A user driven dynamic circuit network implementation. *DANMS08*, 2009.
- [7] HPSS Website. <http://www.hpss-collaboration.org/>.
- [8] J. Li, D. Agarwal, M. Humphrey, C. van Ingen, K. Jackson, and Y. Ryu. eScience in the Cloud: A MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform. In *IPDPS*, 2010.
- [9] X. Qiu, J. Ekanayake, S. Beason, T. Gunarathne, G. Fox, R. Barga, and D. Gannon. Cloud technologies for bioinformatics applications. In *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*, pages 1–10. ACM, 2009.
- [10] M. C. Schatz. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, pages 1363–1369, June 2009.
- [11] J. Thatcher et al. NAND Flash Solid State Storage for the Enterprise - An In-depth Look at Reliability. Technical report, SNIA, 2009.
- [12] From Clusters To Clouds: xCAT 2 Is Out Of The Bag. <http://www.linux-mag.com/id/7230/>.

Appendix A

User Requirements Detail

A.1 Questionnaire

NERSC Magellan - Cloud Computing Interest Detailed Form

We are collecting additional data from users on their applications to understand the needs in cloud environments. Please fill out the form.

* Required

Name *

Email address *

NERSC username

NERSC repo

Title *

Which of the following cloud computing features do you find most attractive. Please check all that apply. *

- Access to additional resources
- Access to on-demand (commercial) paid resources closer to deadlines
- Ability to control software environments specific to my application
- Ability to share setup of software or experiments with collaborators
- Ability to control groups/users
- Exclusive access to the computing resources/ability to schedule independently of other groups/users
- Easier to acquire/operate than a local cluster
- Cost associativity? (i.e., I can get 10 cpus for 1 hr now or 2 cpus for 5 hrs at the same cost)
- MapReduce Programming Model/Hadoop
- Hadoop File System
- User interfaces/Science Gateways: Use of clouds to host science gateways and/or access to cloud resources through science gateways

Please add any other details as to why cloud computing might be attractive to your group.

Application Details

Identify DOE Office and Program of your application *

- Advanced Scientific Computing Research - Advanced Mathematical Sciences
- Advanced Scientific Computing Research - Computer Sciences
- Biological and Environmental Research - Biology Systems Science
- Biological and Environmental Research - Biological Systems Science -SciDAC
- Biological and Environmental Research - Climate and Environmental Sciences
- Biological and Environmental Research - Climate/Environment - SciDAC
- Basic Energy Sciences - Chemical Sciences
- Basic Energy Sciences - Geosciences
- Basic Energy Sciences - Material Sciences
- Fusion Energy Sciences - Fusion Base Program
- Fusion Energy Sciences - Fusion SciDAC
- High Energy Physics - Accelerator Physics
- High Energy Physics - Astrophysics
- High Energy Physics - Lattice Gauge Theory
- High Energy Physics - High Energy Physics Theory

- Nuclear Physics - Accelerator Physics
- Nuclear Physics - Astrophysics
- Nuclear Physics - Lattice Gauge Theory
- Nuclear Physics - Nuclear Theory
- Advanced Networking Initiative (ANI) Project
- Other:

Identify the category that your application falls under. *

- Dense linear algebra
- Sparse linear algebra
- Spectral methods (FFT)s
- Particle Methods
- Structured Grids
- Unstructured or AMR Grids
- MapReduce
- Other:

How do you manage the data for your application? Please check all that apply *

- Use and expect a parallel file system
- Use and expect local disk
- Large amounts of static data is required at the site where application runs
- Input data arrives from offsite
- Output data needs to be pushed out to a different location

Provide rough orders of magnitude on running time of a single application run *

How many runs do you typically perform and how frequently? *

How many cores do you need for a single run? *

Provide rough orders of magnitude on your memory requirements *

Provide rough orders of magnitude of persistent disk requirements *

What is the typical input data size for a single run of your application? * Please enter a number and units of data size

What is the typical output data size for a single run of your application? * Please enter a number and units of data size

What are bandwidth requirements in or out of NERSC? *

Does your application require a specific operating system or version of the operating system? Please specify *

Does your application require specific kernel versions or modules? Please specify *

Does your application require specific libraries? Please specify. *

Would you like to specify any additional details about the application?

Use of Cloud Computing

Describe how many CPU hours you will need for validating cloud computing to be a valid platform choice (proof of concept runs)? *

Describe how many CPU hours you will need for using cloud computing on an ongoing basis? *

Virtual machines are a popular mode of operation in cloud computing that tends to affect some applications in terms of performance. Will this affect your application. Please specify *

- Yes, I anticipate it will run slower but the other features of cloud computing make it more attractive to me
- Yes, it can be a problem
- No
- Maybe
- Don't know

Cloud computing relies on pre-configured images for the OS and software. The nodes do not have any persistent state. Check all that apply that affects your application. *

- My code-base changes often and I need to recompile my code periodically.
- The data is updated often
- I need to save large amounts of data from each run

Will there be other users or collaborators who can use your cloud setup to run their experiments? *

- Yes
- No
- Maybe

Are there others in your scientific community interested in or looking at cloud computing? *

- Yes
- No

Additional comments.

A.2 List of Projects expressing interest

Climate100: Scaling the Earth System Grid to 100 Gbps Networks

PI: Alex Sim(ASCR)

Description:

As one of ANI research projects, we want to use Magellan for research on scaling climate applications to next generation network, computing and storage capacity. We understand that Magellan would not have 100Gbps connection soon. We will use currently available resources first and extend later when higher capability is available. Climate100 is described on <http://sdm.lbl.gov/climate100>.

MapReduce for Bioinformatic tools

PI: Ernest Szeto(BER)

Description:

1. Want to test MapReduce (MR) for BLASTP and BLASTX streaming API. 2. Test higher level scripting tools that don't explicitly require writing MR jobs, such as the data flow (workflow) scripting languages such as PIG. Possibly the same for SQL-like languages (Hbase). 3. Besides batch computing, test data storage tools such as Hbase for low latency interactive UI use. (Use MR for sorting/reorganizing for bulk loading. Use distributed hash for fast retrieval.)

BIO-Portal

PI: Ewa Deelman(BER/Outside)

Description:

This project aims at providing a science gateway for Genomics Workflows. Currently we have a portal <http://wind.isi.edu/wings/index.php?id=53> that provides the ability to run a number of different genomic analysis. We would like to explore the ability to host such an environment on the Magellan Cloud and determine how to support a number of users.

LBL CRD Chombo mid-range and user cloud opportunities

PI: Hans Johansen(ASCR)

Description:

The LBL CRD Applied Numerical Analysis Group (ANAG) would propose using our Chombo software library as a test bed for the appropriateness of cloud platforms for scientific computing. Chombo is a well-established parallel software library for supporting finite difference methods for a variety of scientific computing applications. Chombo's users benefit from adaptive refinement on block-structured grids and highly-flexible cut-cell discretizations. The use of standard open-source tools and libraries allows Chombo to run on desktops, midrange and leadership-class machines.

Objectives of exploring Chombo within Magellan would include: - Evaluating the use of virtual private clusters of 40-1000 cores to support midrange scientific software development on large 2D or small 3D computations, - Investigating Chombo performance on the new platform's architecture (vs. desktop and NERSC machines), and - Hosting external groups of collaborators in an environment that supports small-moderate compute workloads.

Star workflows

PI: Jeff Porter(HEP)

Description:

STAR, a large collaboration running on PDSF, has used cloud services at EC2 to run production scale simulation work. This was implemented by installing a grid resources in a virtualized environment. STAR has also done tests with other virtualization models for production processes. We would like to explore these models further on a dedicated system and, perhaps, leverage STAR resources local to NERSC (PDSF data and NERSC HPSS) to expand cloud usage to more complicated User Analysis processes. Cloud services at

NERSC could potentially provide us with a surge resource which would allow us to use NERSC resources more efficiently.

Cloud Computing for Supernova Cosmology

PI:Rollin Thomas(HEP)

Description:

Mid-range scientific pipeline for spectrophotometric data processing/reduction (Nearby Supernova Factory and Palomar Transient Factory projects). Pipeline currently runs at CCIN2P3 but they no longer fully satisfy our needs: This causes *scientists* to spend time adjusting code to the *imposed* environment rather than discovering things. Usage: Dozens to hundreds of cores, depending on task.

Also VERY interested in science gateways to serve up the data to our collaboration and the public in general; this is a high-profile project to build the next-generation data set for SN cosmology and Dark Energy studies.

Interested in alternative approaches to using computer resources also — rather than using a batch-queue system we are open to something more advanced, but uncertain whether Hadoop or MapReduce would work with our software.

Interested in real-time processing of data from a remote spectrograph also, using a smaller version of the above described pipeline. This would require guaranteed access at specific times, but would only require a few CPUs.

Finally, I have an application for the data-driven analysis of supernova spectra — currently it runs on franklin, but the structure doesn't include a lot of barriers or synchronization (in fact the problem setup is designed to be asynchronous and fault tolerant). This could take advantage of flash memory; but mostly I have simply found that franklin does not really suit my needs. I would like to look into running this code on a cloud computing environment, and also setting up a science gateway so that members of the astrophysics community could submit their own spectra for analysis by my code. Usage: Hundreds of cores.

ATLAS combined interest in Cloud Computing

PI:Yushu Yao & Paolo Calafiura(HEP)

Description:

1. Compare the ATLAS Software performance on a virtualized cluster to the performance on PDSF. Currently the LBL ATLAS group is running analysis jobs on the CHOS based PDSF cluster. ATLAS has both I/O intensive (e.g. data analysis) and CPU intensive (e.g. simulation) jobs. They will provide a good benchmark for testing the performance of virtualized clusters.

2. Test I/O Performance and Reliability of Flash Storage. ATLAS analysis jobs will require reading large amount of data. These data are normally stored in disk arrays and shared for each node in the cluster. Comparing the performance of SSD disk array to conventional disk array will be a valuable study, especially when there is concurrent access of the same storage unit by a number of jobs (hundreds). Study file system optimizations that fit SSD. Besides performance, reliability is another important aspect. Failed disk replacement cost and power consumption are also important factors when making disk purchases.

3. Scalable ATLAS Tier3 Virtual Cluster Many smaller institutions have limited manpower to set up and maintain a batch cluster. We are prototyping a Virtual Tier3 Cluster for ATLAS, where the user (institution) can easily launch a cluster ready for ATLAS jobs. We will need a testbed for testing this prototype in a larger scale, and to measure the performance and scalability of the Virtual Cluster. A lot of research is needed for the infrastructure that provides these services. E.g. security, resource provisioning, usage accounting (cpu, memory, io, storage, remote database access). Another aspect of the same problem is that in many institutions a batch cluster will serve several very different scientific projects (like PDSF). Virtualized clusters can help to reduce maintenance cost and consolidate usage.

4. From Desktop to Cloud Many complex scientific software systems like ATLAS need considerable effort to setup a development or analysis environment. Virtualization has been proven to greatly simplify the deployment of, not only the software itself, but also an working environment for code development and data

analysis. One successful example is the CernVM project. Although development is done on a desktop, jobs are normally sent to the GRID (or Cloud) for execution. A solution to seamlessly integrate the development, deployment and job execution process is needed. This idea can be generalized to how can we use virtualization to deploy a certified work environment to users for the work that needs human input. In this work environment, how are data communicated with the cloud and other work environments. In the cloud, how to handle events initiated from a work environment.

distributed computing utilizing BOINC for study of atmospheric waveguide in approximation of trajectory of cosmic rays

PI:Andrew Gillette(Outside/Astro)

Description:

Scaling of Hadoop to work in a virtual machine environment in distributed processing.

MGTAXA metagenomic binning package on the cloud

PI:Andrey Tovchigrechko(Outside/Bio)

Description:

We have built a cloud image for MGTAXA metagenomic binning package (<http://andreyto.github.com/mgtaxa/>). The initial development of MGTAXA was supported by DOE funds (now it is supported by its own NSF grant). We need to do larger scale testing and also porting from SGE/(share storage) to MapReduce. Ideally the final image will become a publicly available resource on DOE cloud. Part of the NSF grant mandates the creation of web gateway to MGTAXA. Magellan can become one of the backends.

Production Prototype for FSP

PI:Douglas McCune(FES)

Description:

This project will explore the feasibility of setting up a PTRANSP service for time dependent tokamak simulations as a prototype for the Fusion Simulation Program (FSP) capacity production configuration. It is likely that large-scale, high concurrency FSP simulations of full tokamak physics will be preceded by a large number of small-scale simulations of entire discharges at the transport time-scale, using reduced models for some of the fast phenomena. This is especially important for ITER, which requires a thorough scan of all the anticipated plasma parameters and external drivers in order to predict its performance. Our application, the predictive-TRANSP code (PTRANSP), will be typically run using 64 MPI processes for the fast ion model (NUBEAM) and multiple concurrently running ion cyclotron wave heating modules, each one using 96 processes. The total number of MPI processes will depend on the type of discharge being studied although it will remain between 64 to 1024 processes. The hope is to have the users access the cloud resources through a PTRANSP "science gateway".

BIO-Cloud: Exploring the use of Cloud Computing for Genomic Workflows

PI:Ewa Deelman(Outside/Bio)

Description:

This project aims at porting existing and developing new workflow-based applications to the cloud environment. We are currently working with scientists at the USC Medical School to enable their DNA analysis to run on the USC campus cluster. As size of analysis increases (due to the increased amount of data coming of the next generation sequencing machines) the need for additional resources, including those provided by cloud environments increases. As part of this project we propose to port the existing RNA-seq application onto the Magellan cloud. The application consists of several individual steps and can run in approximately 8 hours on 8 cores. We used the Pegasus Workflow Management System to run the application both at USC and on the Amazon Cloud. We would like to compare the performance of the application on HPC resources (at USC) and the different types of Clouds: Amazon and Magellan. Additionally, we plan to develop new workflows that extend the RNA-seq workflow with additional analysis. These are being developed at part

of an NIH Brain Atlas project. We would like to evaluate the performance of these new workflows on HPC and Cloud resources. Depending on the allocation we receive on Magellan, we may also be able to do such comparative studies for other applications, such as those in astronomy or earthquake science. For a list of applications we work with please visit <http://pegasus.isi.edu/applications.php>

Quantum Systems with Cloud Computing

PI: Joseph Carlson(NP)

Description:

We simulate quantum systems in condensed matter, nuclear physics and cold atom physics. Many of these simulations are conducted in a mode of loosely coupled random walks. Cloud computing could certainly be used to perform independent calculations as properties of the system are varied, but we would also like to investigate putting a large-scale simulation into this environment.

The smaller scale simulations are today typically run on local clusters of 100-200 cores. We could compare performance on cloud computing with our local cluster. We are also interested in sharing code/data with collaborators at other universities and national labs.

LUcloud - Cloud enabled LU decomposition for large electromagnetic problems

PI: Peter Messmer(BES)

Description:

Investigations of electromagnetic scattering properties of extended structures are at the core of many scientific and engineering applications.

While a broad range of commercial and free tools exist, most of them do not target large HPC installations. One of the main reasons is that the target users of such facilities, small engineering companies or research departments, do not have access to such installations. Commonly, these users have access to some 128 cores and factorization times of days to weeks are acceptable.

Enabling these researchers and engineers to access high performance computing resources in a cloud has therefore the potential to save tremendous amounts of time.

We are therefore interested in utilizing clouds to provide users of such codes transparent access to high-performance computing platforms. Most of the computational work involves LU factorizations of large, dense linear systems. In some cases, the matrices are so large that out-of-core algorithms need to be used. Being able to utilize some thousand cores instead of a hundred and having access to significantly faster mass storage than regular disks will change the way these scientists can perform their research.

Most of this research will involve workflow investigations (data staging, execution control and monitoring) and some preliminary runs for benchmark studies.

ICECUBE Simulations

PI: Spencer Klein(HEP)

Description:

Icecube is interested in alternative models for access to computing resources, such as use of virtual private clusters reserved for the use of particular research groups for specific periods of time. Icecube workload is unevenly distributed in time and the project could greatly benefit being able to quickly access additional resources and virtualization would alleviate issues created by a fairly complex software environment that makes it difficult to utilize grid resources at this point.

Satellite data processing in the cloud

PI: Deb Agarwal(HEP)

Description:

We are currently using the Microsoft Azure cloud to bring together 6 data products from the MODIS satellite and reprojecting these products onto a 1km grid as needed. Then we calculate evapotranspiration. The input elements are images and the result will be produced as an image (HDF5). We are working with

a scientist that would like to compute ET globally for the period covering the 10 years of data. We would be interested in looking at porting this computation onto the magellan cloud. The problem is embarassingly parallel so it is a good match with the cloud.

Cactus on the Cloud

PI:Erik Schnetter(HEP)

Description:

We develop and support the Cactus software framework that we are also using for the simulation of Gamma-Ray Bursts and other astrophysical systems. Our interest in cloud computing includes increasing the flexibility of supercomputing in several ways:

- We are researching ways in which access to HPC resources can be more flexible, e.g. to support interactive debugging or on-line remote visualisation. Virtual machines, as available on clouds, may be an important step in this direction.

- To increase scalability of future petascale simulations, we are researching ways of splitting off parts of simulations that by construction cannot scale. Presumably these would benefit from running in multiple instances on lower end architectures, e.g. for expensive post-processing stages of individual time steps.

- As some of our astrophysical calculations become less cutting edge, we are beginning to set up a science gateway for binary black hole simulations. Our current intent focuses on teaching due to resource limitations; with proper resources, we would extend this to support astrophysics research, and to include other physics systems.

High resolution simulations of galaxy formation and evolution

PI:Joel Primack(HEP)

Description:

We propose to run 10 different simulations of galaxies with masses similar to the Milky Way. This will give us enough statistics to understand what are the necessary conditions that makes galaxies analogs to the Milky Way. We request a total of 100k cpu-hours in 80 cores (ten nodes) of the new linux cluster, Magellan. This is an example of mid-range scientific computing that uses less than 1k cores. Moreover, each simulation can run independently in a single node. Therefore, this project is well suitable for cloud computing in Magellan. The code uses OPENMP and it needs a shared-memory environment addressing 10-20 Gb of ram per simulation. The same code has been used extensively on bassi, so this project can be very useful for comparative performance tests between bassi and Magellan nodes.

(This project could alternatively use ten nodes on Hopper or Carver. More processors per node would be even better, since the code scales perfectly to 16 processors per node on the Schirra IBM machine at NASA Ames.)

Climate Data Analysis and Management

PI:John Wu, Alex Sim, Lavanya Ramakrishnan(BER)

Description:

The climate data analysis planned require high I/O performance. This I/O requirement is expressed in two forms. It requires transfer of a large amounts of data from remote repositories, and the analysis program typically read a large amount of data from disk into memory. The Hadoop setup can provide very high aggregate I/O throughput, and is well suited for the planned data analysis activity. Some research activities have been reported by our collaborators. Magellan will provide a larger platform for further studying the software system used for parallel data analysis.

The initial software system is planned to use FUSE over HDFS to provide file access to the data analysis algorithms. The FUSE abstraction doesn't let Hadoop use data locality and may result in the analysis computations to be performed on nodes that do not hold data. To provide local file accesses to the analysis code, we plan to update the scheduler of the Hadoop system. We would like to test this prototype on Magellan in the context of climate data analysis. This work will leverage our existing work on FastBit

indexes, and can also benefit a number of other applications running on Magellan system that would benefit from using the Hadoop file system.

The climate data analysis code relies on large data that resides on multiple sites including Livermore, Oak Ridge and NERSC. Eventually the project will also benefit from using the ANI 100G link to transfer data from these sites. This early prototype (with smaller data sets) will help us evaluate the use of technology such as Hadoop file system in preparation for large scale analysis in the future.

Combinatorial electronic structure calculations

PI: Alan Aspuru-Guzik (BES)

Description:

We are interested in exploring HPC clouds as a complement to volunteer-based grid computing, an effort in which we are working on with a considerable part of the Aspuru-Guzik research group (<http://aspuru.chem.harvard.edu>)

THE CLEAN ENERGY PROJECT: Our program will significantly advance our fundamental understanding of materials design rules for the development of polymer-based solar cells. Instead of molecular design by intuition or by experimental synthesis, we use of powerful theoretical techniques to search for ideal chemical compound(s) with the adequate physical and chemical properties needed to generate clean and sustainable forms of energy sources. Working together with IBM, we developed a screensaver (<http://cleanenergy.harvard.edu>), which allows individual users anywhere in the world to contribute their idle computer time to perform electronic structure calculations on combinatorial libraries of molecular compounds suitable for organic solar applications. The deployment of a world-wide distributed computational engine provides an avenue to explore the vast chemical space and identify lead compounds for the next generation of polymer-based solar cells. Currently, advances in the field of computational modeling of materials have brought us closer towards an accurate prediction of the photovoltaic properties of a given molecular material even before experimental synthesis. In this study, we developed a cheminformatics approach that creates molecular libraries using molecular descriptors, and converts them, in a second step, into an initial ensemble of three-dimensional molecular configurations. The resulting Cartesian coordinates are subsequently used to generate input files for the Q-Chem electronic structure software package. The combinatorially created molecular structures and their respective electronic and optical properties as derived from first-principles electronic structure calculations will be used to deploy a database, which will be publicly available to the scientific community for data mining. Ultimately, we hope to achieve a predictive power and provide insight to the experimentalists into the design of molecular architectures that satisfy the high bar for practical devices: air-stable materials, absorption profiles compatible with the solar spectrum, and high charge-carrier transport characteristics.

Using Cloud Computing as a Means for Collaborative Computing

PI: Angelo Raymond Rossi()

Description:

My position is as a Senior Scientist for research computing at the University of Medicine and Dentistry of New Jersey.

In addition to Medical, Dental, and Nursing Schools, etc., there is a Graduate School where researchers are working on important problems in Biomedical research.

My job is to use Computational Chemistry at the intersection of experiment and computation to help researchers understand their results.

I would like to be able to determine how Cloud Computing can be used in a collaborative environment for Computational Chemistry, i.e. "Collaborative Cloud Computing". We have a smaller IBM iDataPlex of 400 cores, and I can use the expertise gleaned from my experience using the Magellan hardware to help researchers to enhance their collaborative abilities through the use of Cloud Computing.

The central thrust of my proposal is to determine whether or not Cloud Computing can be used as an instrument to enhance collaborative research.

Items from the initial list of cloud computing investigations including "science gateways" and easy access to applications could be very useful in "Collaborative Cloud Computing"

Cloud Computing for Electromagnetic Particle-In-Cell Simulations

PI:Arno Candel(BES)

Description:

Comparative performance, especially for tightly coupled codes such as PIC.

Computational Biology, Statistics and Data Analysis

PI:Bill Cannon(Outside/Bio)

Description:

Both application of virtual machines and use of map-reduce programming paradigm for data analysis.

Large Workflow Monitoring

PI:Dan Gunter(ASCR)

Description:

Online monitoring and optimization of large workflows for complex scientific applications.

100G FTP: An Ultra-High Speed Data Transfer Service Over Next Generation 100 Gigabit Per Second Network

PI:Dantong Yu(ASCR)

Description:

100G FTP is newly funded to design and develop an ultra high speed end-to-end file transfer protocol and tool to move science data at a speed of 100 gigabit per second (Gbps) across the national scale 100Gbps data network interconnecting research centers. This project is motivated by the data intensive applications' need for novel data transfer technologies and automated tools that are capable of effectively utilizing available raw network bandwidth and intelligently assisting scientists in replicating a huge volume of data to any desired location in a timely manner. The tool is based on a layered architecture to facilitate a modular design for each component which can be flexibly added and customized in a "plug-and-play" fashion only when needed. The prototype 100G FTP tool will use the advanced Acadia 100G NIC to move data between two DOE Magellan clouds at ANL and NERSC interconnected by the DOE 100Gbps ANI network testbed, and provide the critical data moving layer to two DOE ARRA projects, i.e. Climate 100 and the Advanced Network and Distributed Storage Laboratory. We are interested in using flash storage for data intensive applications. We would like to access to the flash storage that offers substantially increased bandwidth and IOPS (I/O operation rate), and decreased latency. We are also interested in the storage management system, such as, Hadoop Distributed File System (HDFS) that is built on top of cloud computing models such as Hadoop (MAP/Reduce).

Magnetic Fusion Energy Research: Experimental and Simulation

PI:David P. Schissel(FES)

Description:

I manage the Data Analysis Application Group at the DIII-D National Fusion Facility in San Diego. We would like the opportunity to discuss the ideas in more detail. Thanks, David

Experimental: Can cloud computing be used to perform calculations that support an operating tokamak? This calculations are time sensitive, yet require not a large number of processors. Today we run on clusters with up to 70 cores.

Simulation: Clearly cloud computing can help here. Magnetic fusion simulations are of very different size and complexity. Finding one to fit this model should be straightforward.

Comparative performance of scientific applications in cloud computing and traditional cluster/supercomputing environments

PI:Eugene Chen(FES)

Description:

Comparative performance of scientific applications in cloud computing and traditional cluster/supercomputing environments.

Use of flash storage for data intensive applications.

Creation of "science gateways" that use cloud computing to provide easy access to applications, databases or automated workflows.

PI:Gheni Abia(FES)

Description:

I am interested in the creation of "science gateways" that use cloud computing to provide easy access to applications, databases or automated workflows.

Space Weather modelling

PI:Giovanni Lapenta(FES)

Description:

Running space weather services on cloud computing

Application of Cloud Computing for STAR simulation

PI:Jan Balewski(NP)

Description:

Simulation of response of the STAR detector to collisions of high energy proton or heavy ions requires minimal input file, significant CPU per event, and resulting output file is of moderate size. The goal of this study is to assess feasibility of such simulations in the Cloud Computing environment and transfer resulting files to RCF computing facility

Economic viability of cloud computing resources for LBNL midrange computing needs

PI:Krishna Muriki(Outside)

Description:

We would like to run some of the LBNL midrange computational programs on the Cloud computing resources offered by the Magellan project. Applications are picked from various divisions which are currently run on LBNL's institutional cluster namely Lawrenceville. Goal is to do performance comparisons of these selected computational applications in cloud computing and traditional Infiniband cluster environment.

Data storage

PI:Mark Rosenberg(Outside)

Description:

In recent BCP planning it has become clear that tape restore times are too long for file service recovery.

I am looking for a way to mirror our CIFS file server in the cloud. After the disaster I would like to be able to provide direct access to the stored files to Lab staff. Either web or CIFS access would work. There are currently approximately 1000 accounts with 3 TB of data. These represent both home and shared directories. We would need to be able to maintain and enforce the access rights to the data.

Comparative performance of scientific applications in cloud computing and traditional cluster/supercomputing environments.

PI:Massimiliano Alvioli(Outside)

Description:

Comparative performance of scientific applications in cloud computing and traditional cluster/supercomputing environments.

Computational Prediction of Transcription Factor Binding Sites

PI: Mohammed AlQuraishi(BER)

Description:

There are two aspects of our project that may be suitable for cloud computing. First is that the parallelism is typically 1k cores, more in the range of 100s. Second is that much of our analysis involves sequence comparisons, i.e. string-related operations that are particularly suited for cloud APIs.

Impact of RDMA Efficiency on Cloud Computing

PI: Payman Zarkesh-Ha(Outside)

Description:

My main interest is in creating a test bench to measure the impact of RDMA efficiency between servers and storage on cloud computing. More specifically, my focus is on the impact of the latency of high-speed interconnects such as: HyperTransport or QuickPath.

Astrophysics Image Processing

PI: Peter Nugent(HEP)

Description:

I would like to extend the work I did on the SDSC DASH computer with flash storage for data bases and image processing.

National Energy Modeling System

PI: Peter Schwartz(ASCR)

Description:

Our project would attempt to show that a complex program, parallel-NEMS, could be ported to cloud computers. A superfast version of NEMS, would be of great interest. In addition, techniques involved in porting the application would be widely applicable.

REALM

PI: Peter Schwartz(BER)

Description:

The realm project is an initiative between LBNL and the California Department of Water Resources to develop multidimensional models for analyzing flow and water quality in the San Francisco Bay-Delta. Analysis of Bay-Delta management and design alternatives routinely involves numerous long, computationally intense simulations. The potential applications require both MPI-based parallel computation and embarrassingly parallel separate launches representing alternatives or data assimilation replicates. Agencies that study the Bay-Delta (the state, water agencies, consultants and schools) are not able to independently maintain the infrastructure required. We believe that cloud computing would offer an ideal environment, allowing on-demand access to adequate computational capability.

Defects and Charge Transport in Materials for Energy Conversion and Storage

PI: Ram Devanathan(BES)

Description:

We plan to run some data intensive calculations using codes that are moderately parallel. We hope to use this capability for molecular modeling as well as for quick turnaround on data analysis tasks that are not highly parallel. We would like to benchmark the performance against existing resources to evaluate the suitability of cloud computing for our future needs. This will help us evaluate, which types of simulations should be done with massively parallel resources and which are more suitable for cloud computing. Turnaround time and impact on researcher productivity will also be assessed.

Atomic resolution modeling of RNA motifs and protein structures

PI:Rhiju Das(Outside/Bio)

Description:

My lab at Stanford (biochemistry and physics departments) seeks basic principles that will permit a deep and predictive understanding of biomolecule structure. I have extensive experience in distributed computing (Rosetta@home), human computation (FoldIt), and high performance cluster computing through my postdoctoral work in David Baker's lab at UW on the high resolution prediction of protein structures. My new lab at Stanford has recently developed a deterministic, enumerative method that appears to give unprecedented accuracy for small RNA and protein motifs; we are gearing up for CASP10 (the community wide blind prediction trials) this summer of 2010. One of our main efforts is to convert our calculation from being queued by python scripts on Stanford' BioX2 2000-core cluster to the Map/Reduce framework to run on cloud architectures like Amazon EC2. We have run tests on several other clusters, including several TeraGrid systems, and the latency is too high – this seems like a perfect case for a cloud vs. a standard grid-based architecture. We are looking for an academic test-bed that would enable us to test our ideas at low cost; that would inform the work of others searching for new paradigms for scientific computing; and that eventually might be the back-end for a webserver for more general use by the biological community. The NERSC Magellan seems like a superb fit!

Interface of data intensive grid applications to cloud resources

PI:Ruth Pordes, OSG Executive Director; Doug Olson, LBNL liason(NP)

Description:

Groups participating in OSG are investigating various aspects of virtualization and cloud computing and these efforts are especially suitable to cooperation and collaboration with the Magellan project and well positioned to leverage the strengths of the OSG production grid infrastructure with cloud resource models. The STAR experiment, collaborating with the Nimbus group at ANL, has utilized Amazon EC2 for a science production run by deploying a virtual OSG compute element with several hundred cores as well as testing other models of virtual resources at Clemson and Wisconsin. The Panda pilot-job framework is currently providing a production science gateway for the ATLAS experiment and other science groups on OSG. There is an effort to investigate an interface to virtualized resources that is interested in collaboration with Magellan. The CMS experiment is also investigating technical issues and cost models for cloud computing, with scientists at Vanderbilt testing the usability of the Monte Carlo production software stack. The group at Clemson is investigating cloud aspects of deploying virtualized resources on a campus grid and can provide valuable insights and examples of university researchers interfacing with a scientific cloud facility. The Condor project has been interfaced to Amazon EC2 and developing capabilities in managing virtual machines is very interested in adapting capabilities suitable for Magellan. The OSG Engagement group is working with a variety of applications, some of whom may benefit from testing on virtualized cloud resources. Given these interests by OSG groups and the OSG expertise in deploying a distributed infrastructure OSG is very interested in collaborating on investigating the issues of interfacing grid services to cloud resources and especially looking at data flows between the NERSC and ANL resources utilizing the ANI network. The Advanced Network and Distributed Storage Laboratory (ANDSL) for Data Intensive Science is interested in using the Magellan endpoints to deploy distributed storage solutions supported by OSG, and verify their capabilities once the 100Gbps testbed becomes available.

Automatic Calibration of Astronomical Images

PI:Sam Roweis(Outside/Astro)

Description:

Use of distributed storage and computing to support a system which allows amateur scientists to contribute data to a public archive which is robotically calibrated to ensure its scientific reliability.

Run climate models

PI:Shenjian Su(BER)

Description:

Try to run climate models in Cloud Computing mode.

Performance Evaluation of On-Demand Provisioning of Data Intensive Applications

PI:Sriram Krishnan(Outside)

Description:

We would like to perform a detailed benchmarking effort for data intensive applications using the resources provided by Magellan. This work will complement our work being performed on private and public cloud-based environments using the NSF CluE facility and Amazon Web Services (AWS). We have already received an allocation of resources under Amazon's Educational program to perform this study. The benchmarking work will focus on high resolution topographic datasets and tools that are provided by the OpenTopography project (<http://www.opentopography.org>) at SDSC. Under the scope of this project, we wish to deploy DB2 and Hadoop on Magellan resources; install our benchmark data sets; and execute a number of benchmark queries on these data sets, using both DB2 and Hadoop-based solutions for serving these data sets. In addition to delivering the benchmarking results, this work will also lead to working out the "teething" troubles of running parallel DB2 as well as the Hadoop parallel execution environment across the Magellan resources. The goal of this project is to investigate whether a resource such as OpenTopography can be successfully hosted on cloud-based resources. We anticipate that if we are to host OpenTopography "on the cloud", we will need a high speed network connection between SDSC and the cloud (Magellan, in this case).

Use of flash storage for data intensive applications

PI:Wei-keng Liao(ASCR)

Description:

Parallel I/O, MPI-IO optimizations. Energy consumption optimization for hard-drive and flash-drive hybrid storage systems.

Exploring cloud computing environment for fast biological sequencing applications

PI:Weikuan Yu(Outside)

Description:

Long-distance I/O, Map/Reduce, Flash storage

Transfer HPC applications to Cloud platform

PI:Yong Qin(FES)

Description:

We are trying to understand how the Cloud technology can benefit us. Thus we are trying to transfer our current HPC applications to the Cloud platform and/or redesign algorithms to adopt the new platform.

Appendix B

Relevant Publications

Defining Future Platform Requirements for e-Science Clouds

Lavanya Ramakrishnan *
lramakrishnan@lbl.gov

Keith R. Jackson *
krjackson@lbl.gov

Shane Canon +
scanon@lbl.gov

Shreyas Cholia +
scholia@lbl.gov

John Shalf +
jshalf@lbl.gov

* Advanced Computing for Science
Lawrence Berkeley National Lab
Berkeley, CA

+ NERSC
Lawrence Berkeley National Lab
Berkeley, CA

ABSTRACT

Cloud computing has evolved in the commercial space to support highly asynchronous web 2.0 applications. Scientific computing has traditionally been supported by centralized federally funded supercomputing centers and grid resources with a focus on bulk-synchronous compute and data-intensive applications. The scientific computing community has shown increasing interest in exploring cloud computing to serve e-Science applications, with the idea of taking advantage of some of its features such as customizable environments and on-demand resources. Magellan, a recently funded cloud computing project is investigating how cloud computing can serve the needs of mid-range computing and future data-intensive scientific workloads. This paper summarizes the application requirements and business model needed to support the requirements of both existing and emerging science applications, as learned from the early experiences on Magellan and commercial cloud environments. We provide an overview of the capabilities of leading cloud offerings and identify the existent gaps and challenges. Finally, we discuss how the existing cloud software stack may be evolved to better meet e-Science needs, along with the implications for resource providers and middleware developers.

Categories and Subject Descriptors

C.2.4 [Computer Systems Organization]: Computer Communication Networks—*Distributed Systems*; J.2 [Computer Applications]: Physical Sciences and Engineering

General Terms

Design, Performance

Keywords

cloud computing, data parallel computing, mapreduce, scientific computing

Copyright 2010 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. SoCC'10, June 10–11, 2010, Indianapolis, Indiana, USA. Copyright 2010 ACM 978-1-4503-0036-0/10/06 ...\$10.00.

1. INTRODUCTION

Cloud computing provides a new resource model where multiple virtual servers hosted in data centers are used by individuals or groups, usually through a *pay-as-you-go* model. Cloud computing provides an illusion of infinite computing resources available on demand, i.e. in current cloud systems resources are accessible to the user almost instantly, with startup time of the instance imposing the only delays.

Cloud computing platforms are primarily used to serve the needs of web 2.0 applications, whereas their use in scientific communities is still being evaluated [9]. Different e-Science groups including those in bioinformatics [13], astronomy [5] and high energy physics have experimented with Amazon's infrastructure-as-a-service model and have also investigated the use of Hadoop for programming loosely coupled applications. Early results indicate some performance degradation in comparison to conventional batch-scheduled clusters. But this also promises to be an avenue to address new categories of scientific applications including data intensive science applications, on-demand/surge computing, and applications that require customized software environments. This new resource model will have a substantial impact on the business model for future High Performance Computing (HPC) centers in terms of how they provide services to the scientific community and the evolution of the software infrastructure necessary to manage those resources.

Magellan is a recently funded project, through DOE ASCR, to investigate how the cloud computing business model can be used to serve the needs of midrange computing and future data-intensive computing workloads for the Office of Science that are not served through DOE data center facilities today. The distributed testbed infrastructure has been deployed at Argonne Leadership Computing Facility (ALCF) and the National Energy Research Scientific Computing Facility (NERSC). At NERSC, the testbed will consist of 1,440 Intel Nehalem quad-core processors (5,760 cores total).

There has been intense discussion on the characteristics of clouds, along with their benefits and comparisons with grid environments [1, 7]. However, comparatively little attention has been devoted to defining the workload requirements, business model, user management, scheduling, application tools and security in the context of e-Science applications. The latter topics are central to defining the role of cloud computing in existing supercomputing centers and DOE's investment strategy for future computing infrastructure.

Cloud computing encompasses a wide scope of technologies and offerings. In this paper, we specifically address the solutions that are pertinent to e-Science applications - use of virtualized environments and software tools that are useful in these environments.

Section 2 defines a taxonomy of key scientific workloads that might be served well through cloud environments based on their application characteristics and associated business model. Section 3 describes examples of e-Science workloads that could benefit from existing cloud technologies. Section 4 identifies gaps in the current cloud computing offerings for target workloads. Section 5 revisits the cloud service model and outlines the opportunities of an improved software stack that better meets the needs of e-Science applications.

2. E-SCIENCE ENVIRONMENTS

Scientific explorations consist of a broad spectrum of application codes that run from user desktops to supercomputing centers, in areas such as nuclear physics, bioinformatics, environmental sciences, etc. These applications have a varied set of requirements and often have a need for unlimited compute cycles and data storage.

2.1 Application Classes

In this section, we provide a high-level classification of workloads in the scientific space based on their resource requirements and delve into the details of why cloud computing is attractive to these application spaces.

Bulk-Synchronous Large-Scale Computations. These are complex scientific codes generally running on large-scale supercomputing centers across the nation. These are MPI codes using a large number of processors (often in the order of thousands) and may have long running jobs. These jobs are serviced at supercomputing centers through batch queue systems. Users wait in a managed queue to access the resources requested, and their jobs are run when the required resources are available and no other jobs are ahead of them in the priority list. Most supercomputing centers provide archival storage and parallel file system access for the storage and I/O needs of these applications.

Bulk-Synchronous mid-range. These applications run at a smaller scale than the above jobs. There are a number of codes that need tens to hundreds of processors. Some of these applications run at supercomputing centers and backfill the queues. More commonly, users rely on small compute clusters that are managed by the scientific groups themselves to satisfy these needs.

Asynchronous massively independent. Some scientific explorations are performed on the desktop or local clusters and have asynchronous, massively independent computations. Even in the case of large-scale science problems, a number of the data pre- and post-processing steps, such as visualization, are often performed on the scientist's desktop. However the increased scale of digital data due to low cost sensors and other technology [10] has resulted in the need for these applications to scale to environments such as cloud environments. The requirements of such applications are similar to those of the internet applications that currently dominate the cloud computing space, but with far greater data storage and throughput requirements.

The Integrated Microbial Genomes (IMG) system hosted at the DOE Joint Genome Institute (JGI) fits this category.

It supports analysis of microbial community metagenomes in the integrated context of all public reference isolate microbial genomes. The content maintenance cycle for data involves running BLAST for identifying pair-wise gene similarities between new metagenome and reference genomes where the reference genome baseline is updated with new (approximately 500) genomes every 4 months. This processing takes about 3 weeks on a Linux cluster with 256 cores. The size of the databases is growing, it is important that the processing can still be accomplished in a timely manner. The primary computation in the IMG pipeline is BLAST, a data parallel application that does not require communication between tasks and thus has similarities with traditional cloud applications.

2.2 Usage model

A central component of cloud computing is the underlying usage or business model. Currently, midrange computing infrastructure consists of a large number of departmental or PI (Principle Investigator)-owned clusters that are distributed across the DOE - some placed in machine rooms and many housed in closets. The virtualization technology that enables the cloud computing business model to succeed for Web 2.0 applications, could be used to create virtual "private clusters" within a shared resource that look, for all practical purposes, to be identical to privately managed PI-owned clusters. The premise is that carving up machines from large-scale data centers enables substantial cost-savings due to economies of scale and improved energy efficiency compared to running a number of smaller clusters, while retaining all of the benefits of exclusive control of the software configuration and availability that PIs desire.

The model thus facilitates the outsourcing of resource needs to external providers on a *pay-as-you-go* model instead of maintaining local infrastructure. A number of major vendors including IBM and HP, and more recently Amazon and Microsoft have embraced this service model of operating large clusters on behalf of external clients. In the context of the application classes discussed above we describe relevant usage models.

Private cluster. Some scientific users prefer to run their own private clusters for a number of reasons. They often don't need the concurrency levels achievable at supercomputing centers and need guaranteed access to resources for specific periods of time. For these needs to be satisfied in cloud environments, we need to be able to provide guaranteed access to the cluster when needed. This matches the level of service that motivates them to operate their own private cluster.

Personalized environment. A number of scientific applications have strong OS version dependencies and need environments that are consistent with local cluster or desktop environments. In these cases while users might not care where the resources are located, they desire the flexibility associated with custom software environments. The instant-availability of the resource is not as critical to this class of users as the strict control of the entire software environment (down to the sub-revisions of the OS kernel and libraries), and the throughput of the solution.

Science Gateways Users of well-defined computational workflows often prefer to have simple web-based interfaces to their application workflow and data archives. Web interfaces enable easier access to resources by non-experts, and enable

wider availability of scientific data for communities of users in a common application area (e.g. Virtual Organizations). In this case the underlying infrastructure is decoupled from the user interface - scientists interface directly with their workflows through the web. Applications become available to users using the Software-as-a-Service model.

2.3 Application Requirements

Each of the current computational modes available to e-Science applications has certain disadvantages that cloud computing promises to address. We provide a brief overview here of the needs of the applications.

Scalable Computational Capacity The scientific exploration process often requires a large number of runs with different parameters and configurations. Supercomputing centers are shared resources that often have long queue wait times and policies on how many jobs a user might have in the queue. In addition, users often have bursts in their resource needs (e.g. close to deadlines), that might be unpredictable. Such special access needs often require out-of-band discussions with resource providers and a number of such requests are not accommodated due to over-subscription of the resources. Cloud computing promises the ability to get access to unlimited resources (albeit at a cost) which is very attractive to scientific users who have periodic surges in their resource needs that can't be satisfied through local or high performance computing centers.

Scalable Computational Performance. Many distributed memory scientific applications depend heavily on message passing libraries (MPI or GasNET) for inter-processor communication. In particular, bulk synchronous applications are very sensitive to messaging latency and bandwidth. In order to reduce latency and improve bandwidth, MPI-oriented HPC systems typically use high-performance fabrics like Infiniband and use messaging technology that bypasses the operating system to give direct access to the hardware. Collective constructs in MPI applications, such as barriers and reductions, are highly sensitive to subtle load-imbalances. Therefore reducing random sources of noise in the operating system (OS jitter) is critical for maintaining scalable performance.

Consistent Software Environments. HPC systems and, often, local cluster resources are shared environments where users are affected by any software upgrades that might occur at the sites. Compiler and library upgrades can cause many unproductive hours for the scientist. Currently, users either have to spend hours upgrading their software environments or must explicitly run only on resources where the environment is compatible with their needs. Thus user groups are often unable to use available idle resources during periods of surge due to software incompatibility.

Software distributions. A number of large-scale multi-institution collaborations have common shared software codes. The source code for these applications are distributed today and each group then individually installs them on local resources. Due to the variability in the software supported at each site, configuring and reproducing the exact execution state requires hours to days of work and coordination. Some groups are investigating the use of virtual machine images for distribution of all required software [3]. This would enable sites to boot the virtual machines at different sites with minimal or no work involved with software management.

Programming model. Scientific computing facilities

mostly serve the needs of high-end compute intensive scientific codes. But as digital data becomes more readily available, there is an increasing need for scaling data intensive science that has traditionally run on desktop machines. Cloud computing works well with the MapReduce programming model that promises to be useful for some applications with a large amount of parallel data processing. For example, the JGI IMG pipeline mentioned earlier can benefit from a MapReduce framework where a number of sequence comparisons are processed in parallel.

Data Storage and Data Management HPC systems typically provide high-performance parallel file systems that enable parallel coordinated writes to a shared filesystem with high bandwidth and capacity. The file system also needs to allow for access patterns where multiple clients concurrently write to the same file, which is not supported by NFS based solutions. Parallel file systems like GPFS, Lustre and Panasas are usually employed to meet these needs and are often layered on top of high-end storage systems or use specially designed hardware. Even with these capability focused file systems and storage systems, data storage and movement remain one of the most challenging aspects of high-performance computing. In addition to the high-performance file systems, HPC centers typically provide an archival storage system to archive critical output and results. These systems are geared towards storing many petabytes of data in a reliable fashion. Many of these systems still use tape for storage. This is both for its cost effectiveness per byte and reliability. Cloud computing models will need similar data storage and management options for scientific applications to effectively use these environments.

3. CURRENT CLOUD SYSTEMS

We have worked with a number of applications and experimented with different cloud providers and technologies available. We have run a number of high performance benchmark [8] and application codes on Amazon web services to understand the performance implications of virtual machines. In addition, we have used Hadoop to manage the BLAST computations in the JGI IMG pipeline and compared its performance on different platforms including traditional HPC platforms, Amazon EC2 and the Yahoo M45 clusters [2]. In these sections we detail our experiences with using these technologies for these application studies.

3.1 Hadoop

The Apache Hadoop project is an open-source software that provides capabilities to harness commodity clusters for distributed processing of large data sets through the MapReduce [4] model.

The Hadoop streaming model allows one to create map and reduce jobs with any executable or script as the mapper and/or the reducer. This is the most suitable model for scientific applications that have years of code in place capturing complex scientific processes. The Hadoop framework does, however, make assumptions about the data model (e.g., single line inputs per process) that are not valid for scientific applications. This requires re-engineering of the application data used with Hadoop jobs.

The Hadoop File System (HDFS) is the primary storage model used in Hadoop. HDFS is modeled after the Google File system and has several features that are specifically suited to Hadoop / MapReduce. Those features include ex-

posing data locality and data replication. Data locality is a key aspect of how Hadoop achieves good scaling and performance and Hadoop attempts to locate computation close to the data. This is especially true in the Map phase which is often the most I/O intensive phase. Data is replicated for fault tolerance and to provide more opportunities to execute computation near data. The data is transparently replicated by the file system.

3.2 Amazon Web Services

Amazon Web Services is a very popular cloud computing platform today. Amazon provides a number of different instance types in terms of its computational power for different pricing. We have run a number of benchmarks on the platform and the high performance MPI applications tend to experience a performance hit. Additionally, Amazon also provides higher-level services such as Elastic Map reduce. However our experience with elastic map reduce revealed that applications that didn't fit the traditional Hadoop data model could not use the existing API.

The primary methods for data storage in Amazon EC2 are S3 and Elastic Block Storage (EBS). S3 is a highly scalable key based storage system that transparently handles fault tolerance and data integrity. EBS provides a virtual storage device that can be associated with an Elastic Computing instance. S3 charges for space used per month, the volume of data transferred and the number of metadata operations (in 1000 allotments). EBS charges for data stored per month.

Scientific experimentation often results in changes to code and configuration, which may involve recreation of the virtual machine image. To avoid that problem and to additionally serve as the global file system needed for MPI jobs, we used EBS on a single node containing our binary and input data. This was then mounted through NFS on the rest of the nodes.

3.3 Yahoo M45

The Yahoo M45 cluster is a shared Hadoop platform-as-a-service cloud environment. The cluster resources are shared amongst users using a fair-share scheduler. The goal of this cluster is to provide a Hadoop MapReduce platform for data parallel applications in the scientific research space. At the time of writing, the cluster is comprised of 400 dual quad-core Intel Xeon E5320 1.86GHz nodes with 6GB of memory per node. Each node is configured to run 2 map tasks and 1 reduce task.

We did a performance analysis for the IMG metagenomics computations using the BLAST application in this framework and found that it was comparable to that of other traditional HPC platforms and Amazon EC2. Memory appears to be the bounding factor for BLAST, since the genome database must be loaded in memory. BLAST itself scales linearly with the size of the database but we hit a performance cliff when the database size exceeds the available memory size. Additionally, since our tests were run on a shared Hadoop cluster, it was difficult to get consistent overall performance numbers. The load on the system from other users impacted our total time-to-solution.

The Yahoo! M45 system allowed us to identify important bottlenecks and limitations in this problem space. In order to make the M45 environment more suited to a cloud-implementation of BLAST or similar memory intensive scientific applications we would benefit from a) higher memory

limits in the software configuration, b) nodes with a large amount of available physical memory (similar to the extra-large instances on Amazon EC2 which have 15GB) c) ability to perform reservations for on-demand computing giving us more reliable throughput times for total time-to-solution.

4. GAP ANALYSIS

In the previous section, we detailed current cloud offerings. Here we examine some of the gaps and challenges in using existing cloud offerings directly for scientific computing.

4.1 Resource Provider Policies

Clouds promise an unlimited supply of resources on-demand. While this was true in early days of cloud computing where demand for resources was still ramping up, more recently users have noticed that their requests have not been satisfied on providers such as Amazon EC2 due to insufficient capacity. This situation is similar to current day supercomputing and grid resources that are often over-subscribed and have long wait queues. Thus for the end-scientist cloud computing as an unlimited supply of cycles tends to be less promising. There is a need for differentiated levels of service similar to Amazon's current offerings but with advanced resource request interfaces with specific QoS guarantees to avoid users needing to periodically query to see if resources have become available.

Portability. The vision of grid computing has been to enable users to use resources from a diverse set of sites by leveraging a common set of job submission and data management interfaces across all sites. However experiences revealed that there were challenges due to different software stacks and software compatibility issues. Virtualization facilitates software portability. Open source tools such as Eucalyptus [11] enable transition from local sites to Amazon EC2, but cloud interfaces in general are diverse and specific to each site making it hard to easily use multiple sites. In addition, cost of data movement to and especially from the cloud tends to be very expensive, discouraging portability. The data costs are also an issue for applications where scientists would like to perform post-processing on their end-desktop or local clusters, or would like to share their output data with other colleagues.

Cost. Today's cloud providers have a pay-as-you-go model for cloud resources. Cloud computing essentially enables anyone with a credit card to get access to resources. However infrastructure needs for scientific computing are either addressed through large upfront grants for equipment and/or through peer-reviewed allocations on supercomputing resources. Credit card transactions for resources don't fit into the current budget model at research institutions. In addition, PIs often have to distribute or carve out some percentage of their entire allocation to different users which cannot be accomplished in today's cloud computing scenarios. In addition, providers such as Amazon EC2 provide a plethora of options (e.g., spot instances, reserved instances, etc). under different pricing models. The diversity and unlimited scope of the scientific processes necessitates a runtime cost-benefit evaluation with respect to these offerings. Thus, we need to revisit institutional policies and software frameworks that can capture these policies to spearhead cloud computing adoption for e-Science.

4.2 Application Performance

Scientific applications have fairly large memory, compute, data and network needs. Our experiences with running Blast on the Yahoo! M45 cluster pushed us against some of these limits. For example, Blast performance scales linearly with the size of data, and is bound by the available physical memory - once the search database can no longer fit into physical memory we notice a sharp performance drop-off.

The traditional synchronous applications such as MPI perform poorly on virtual machines and have a huge performance overhead. Application codes with minimal or no synchronization, modest I/O requirements, with large messages or very little communication tend to perform well in cloud virtual machines. Traditional cloud computing platforms are designed for applications where there is little or no communication between individual nodes and where applications are less affected by failures of individual nodes. This assumption breaks down for large-scale synchronous applications.

In general cloud providers need to build systems that can meet the more intensive requirements of scientific computing. The standard commercial offerings might not be completely suitable for scientific needs out of the box, and at the very least, seem to require a fair amount of initial setup to meet the needs of science applications.

Cloud providers such as Amazon EC2 provide simple web service APIs for access and setup of resources. However running on Amazon EC2 often requires creating customized images, determining how resources are managed, implementing fault tolerance, etc. This requires a fair amount of system administrator experience. Similarly Hadoop applications require a fair amount of development experience. Thus there is a need for developing application tools that exist above these current offerings that account for the needs of the science and enable easier access to cloud resources.

Additionally in the cloud model, time-to-solution is a more critical metric than individual node performance. Scalability is achieved by throwing more nodes at the problem. However, in order for this approach to be successful for e-Science, it is important to understand the setup and configuration costs associated with porting a scientific application to the cloud - this must be factored into the overall time-to-solution metrics in resource selection and management decisions.

4.3 Data management

As discussed earlier, scientific applications have a number of data and storage needs. Synchronized applications need access to parallel file systems. There is also a need for long-term data storage. None of the current cloud storage methods resemble the high-performance parallel file systems that HPC applications typical need. Hadoop is optimized such that applications can benefit from data locality in the underlying HDFS system. This requires that Hadoop applications are written to be able to store and retrieve data from HDFS. Thus while applications can leverage the features of Hadoop for task farming and coordination of tasks, rewriting legacy scientific applications for use in cloud environments is often infeasible and impractical. File system modules have been written for Linux that allow access to HDFS through the standard VFS layer. The FUSE interface takes VFS request from the kernel and executes them in user space and this results in a performance overhead.

5. REVISITING THE CLOUD MODEL

The goal of the Magellan project is to evaluate possible solutions for cloud computing for science. In this section, we revisit the challenges and opportunities facing resource providers and middleware developers in providing next-generation services.

Resource Provider Model. Resource providers serving the e-Science community need to support a diverse set of models to accommodate different user needs. For example, synchronous applications often need access to non-virtualized resources for the best performance. Similarly as sites support more data intensive science, there is a need for frameworks such as Hadoop. Magellan will support provisioning through batch queue (MOAB) systems, cloud solutions such as Hadoop, and Eucalyptus, as well as access to Hadoop over batch queue systems using Hadoop On Demand (HOD).

Virtual machines do not have any persistent state or shared file systems, requiring sites to develop solutions that meet the storage needs of the end user. Resource provider sites need a) shared file systems such as GPFS or NFS accessible across all nodes allocated to the end user, that can be used to store persistent information across virtual machines and b) archival storage for long term storage of data.

Currently in supercomputing centers, the sites manage the operating system and middleware that is needed across multiple groups. Users compile and install their applications on specific systems (often with help from site personnel). As we move to the cloud computing model, sites must provide tooling and support for managing a diverse set of kernels and operating systems that required by specific groups. The clear separation of responsibilities for software upgrades and operating system patches no longer exists, and sites will need mechanisms to bridge the gap between supporting user-supported images and site security policies.

A cloud system where users have complete control of their images, has certain implications on site security policies. Additional checks and mechanisms are needed to protect critical infrastructure services (e.g. DNS, file servers) on the sites, ensure isolation between different virtual machines, and defend against malicious outgoing and incoming traffic.

We need monitoring and dynamic allocation policies that can load-balance between clusters of different types and provide guaranteed Quality of Service to users of all resource types. In addition to the performance needs of the application, there is a need for tools to manage fault tolerance and reliability of the virtual machines. Virtual machine migration [12] has been proposed as a way to provide higher quality of service especially during planned upgrades and maintenance cycles.

Software Stack. Software stacks supporting distributed scientific applications has largely evolved in the context of grid computing. The software stack can be categorized into these primary layers a) End-user portal interfaces or science gateways b) Coordination services layer that coordinates underlying resources for efficient and reliable execution c) Resource management services that interface directly with the underlying resources

Cloud computing addresses the portability of the software stack, a known issue with current day grid systems. Cloud computing for e-Science needs a similar set of software tools to harness and coordinate underlying resources. Cloud computing technologies provide additional features that greatly

simplify some of the known challenges with existing software stacks, and also provides additional challenges of resource coordination. Here we present a vision for a cloud computing software stack for e-Science that mitigates some of the known problems and builds on existing commercial products.

Batch queue systems such as Moab, PBS, etc provide customizable policy points and algorithms to control scheduling of requests and resources. Similarly, there is a need for policy points in the cloud infrastructure and the ability to store and service requests when resources cannot be allocated immediately. On top of the basic resource management services, there are a number of coordination services available in cyber-infrastructure environments today. These include grid services for job submission and data transfer, application services that coordinate the application specific setup and execution, data replication services, meta-scheduling services etc. In cloud computing we need a number of similar tools and services. These tools must manage the underlying resource procurement, and apply any runtime customization needed in the virtual machine (e.g. bringing up specific services within the instance). We need monitoring services and the ability to dynamically grow and shrink resource holdings without impacting the application. Additionally, we need tools to manage job and data coordination, execution and monitoring.

Moving higher up in the software stack, a class of users will expect to interface with their underlying applications through web-based science gateways, so that they are removed from the infrastructure level details. Applications must then be accessed through a middleware layer that can expose traditional science applications as web services.

In order to make scientific computing more accessible, and to allow for a faster overall time-to-solution, it is important to have simple and robust interfaces that are seamlessly integrated with the applications and cloud infrastructure. Given the nature of the cloud, and its close coupling with web technologies, these interfaces must leverage existing http based protocols. REST [6] (Representational State Transfer) is gaining wide adoption as the underlying mechanism for building these interfaces. REST allows one to expose resources by using the existing http protocol layer. It creates a very simple and powerful means to access underlying resources using a combination of URIs and http verbs.

Most cloud providers including Amazon, Microsoft, Google, etc., are striving to make their interfaces RESTful. Grid technologies, which are similar in spirit to the cloud have been saddled by more heavyweight protocols and technologies like WSDL and SOAP, limiting overall usability. Using simple REST based APIs makes it very easy to add powerful web 2.0 functionality to science applications. Looking forward, this will allow users to *mash-up* disparate data sources across the cloud because everything is essentially just a URI.

6. CONCLUSIONS

Cloud computing promises to be an alternative approach for midrange and data intensive e-Science applications. Scalable parallel performance is achievable on these platforms for application codes with modest I/O requirements and minimal or no synchronization and communication. However, existing cloud offerings in the commercial space do not completely meet the needs of these applications. Currently,

running a particular application on the cloud requires substantial understanding and assembly of the cloud computing technologies. This paper summarizes these gaps and revisits the service model both in the context of providers as well as the software stack. There is a need for differentiated service levels, hardware platforms tailored to scientific applications, and higher level software tools that can manage the complexities of the underlying technology fabric.

7. ACKNOWLEDGMENTS

This work was funded in part by the Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-C02-05CH11231. The authors would like Yahoo! and CITRIS, UC Berkeley for access to Yahoo! M45 cluster and Greg Bell and CITRIS for Amazon EC2 access. The authors would also like to thank the Magellan team at NERSC for discussions on cloud computing, Victor Markowitz for detailed discussions on the IMG pipeline.

8. REFERENCES

- [1] M. Armbrust and et al. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
- [2] S. Canon, S. Cholia, J. Shalf, K. Jackson, L. Ramakrishnan, and V. Markowitz. A performance comparison of massively parallel sequence matching computations on cloud computing platforms and hpc clusters using hadoop. 2009.
- [3] Cern Virtual Machines. <http://rbuilder.cern.ch/project/cernvm/releases>.
- [4] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. pages 137–150.
- [5] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The Cost of Doing Science on the Cloud: The Montage Example. In *Proceedings of SC'08*, Austin, TX, 2008. IEEE.
- [6] R. T. Fielding and R. N. Taylor. Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2):115–150, 2002.
- [7] I. Foster, Y. Zhao, I. Raicu, and S. Lu. Cloud Computing and Grid Computing 360-Degree Compared. *Grid Computing Environments Workshop, 2008. GCE '08*, pages 1–10, Nov. 2008.
- [8] J. S. Katie Antypas and H. Wasserman. Nersc-6 workload analysis and benchmark selection process. Technical Report LBNL-1014, Berkeley, CA, 2008.
- [9] K. Keahey and T. Freeman. Science Clouds: Early Experiences in Cloud Computing for Scientific Applications. In *Cloud Computing and its Applications (CCA)*, 2008.
- [10] J. Li, D. Agarwal, M. Humphrey, C. van Ingen, K. Jackson, and Y. Ryu. eScience in the Cloud: A MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform.
- [11] D. Nurmi and et al. Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems. Technical Report 2008-10, University of California, Santa Barbara, California, August 2008.
- [12] K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe. Live data center migration across wans: a robust cooperative context aware approach. In *INM '07: Proceedings of the 2007 SIGCOMM workshop on Internet network management*, pages 262–267, New York, NY, USA, 2007. ACM.
- [13] M. C. Schatz. CloudBurst: highly sensitive read mapping with MapReduce. *Bioinformatics*, pages 1363–1369, June 2009.

Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud

Keith R. Jackson
and Lavanya Ramakrishnan
Advanced Computing for Science
Lawrence Berkeley National Lab
Berkeley, CA 94720
KRJackson@lbl.gov
LRamakrishnan@lbl.gov

Krishna Muriki
Information Technology
Lawrence Berkeley National Lab
Berkeley, CA 94720
KMuriki@lbl.gov

Shane Canon, Shreyas Cholia, John Shalf
Harvey Wasserman, and Nicholas J. Wright
NERSC
Lawrence Berkeley National Lab
Berkeley, CA 94720
SCanon@lbl.gov, SCholia@lbl.gov, JShalf@lbl.gov
HJWasserman@lbl.gov, and NJWright@lbl.gov

Abstract—Cloud computing has seen tremendous growth, particularly for commercial web applications. The on-demand, pay-as-you-go model creates a flexible and cost-effective means to access compute resources. For these reasons, the scientific computing community has shown increasing interest in exploring cloud computing. However, the underlying implementation and performance of clouds are very different from those at traditional supercomputing centers. It is therefore critical to evaluate the performance of HPC applications in today’s cloud environments to understand the tradeoffs inherent in migrating to the cloud. This work represents the most comprehensive evaluation to date comparing conventional HPC platforms to Amazon EC2, using real applications representative of the workload at a typical supercomputing center. Overall results indicate that EC2 is six times slower than a typical mid-range Linux cluster, and twenty times slower than a modern HPC system. The interconnect on the EC2 cloud platform severely limits performance and causes significant variability.

I. INTRODUCTION

Cloud computing has emerged as an important paradigm for accessing distributed computing resources. Commercial providers such as Amazon, Rackspace, and Microsoft, all offer environments for developing and deploying applications in the cloud. Several groups have reported studies of the applicability of these environments for scientific computing. While it has been shown that these environments are suitable for some classes of scientific applications, it is equally important to understand performance for the broader array of applications that currently run at major supercomputing centers. This is because cloud computing, in one form or another, has been suggested as a potential platform for supporting mid-range scientific computing workloads that play a vital and growing role in basic research across all scientific disciplines [1].

This paper explores the performance of a commercial cloud environment, the Amazon Elastic Compute Cloud (EC2) [2], by analyzing a series of high-performance computing (HPC) application benchmarks that represent a broad cross-section of a typical workload at a major supercomputing center.

While building on our previous efforts [3], [4], [5], [6] and complementing several other related studies [7], [8], [9], [10], the unique contribution of this work is that it presents the broadest evaluation to date of application performance on cloud computing platforms. Our application suite includes a diversity of numerical methods and data-structure representations in the areas of climate, materials science, fusion, accelerator modeling, astrophysics, and quantum chromodynamics. In addition to simply recording the performance of the applications we also instrument the runs using the Integrated Performance Monitoring (IPM) [11] framework. This allows us to determine, in a non-perturbative manner, the amount of time an application spends computing and communicating using MPI, which provides insight into which aspects of the underlying architecture are affecting performance the greatest. Additionally, our approach includes a well-documented method for summarizing achieved, application-level performance based on a simple aggregate measure that expresses useful potential of the systems considered.

Our evaluation compares a diverse set of platforms that range from fully-integrated HPC systems such as a Cray XT4, to two commodity Linux/Infiniband clusters, to the commercial Amazon EC2 cloud environment. These systems cover a spectrum of component customization, integration, energy efficiency, and other characteristics that arise from varying workload orientations.

We have chosen the Amazon EC2 web service as a cloud platform because of its popularity, and because it is broadly representative of mainstream cloud environments that are based on some form of hardware virtualized machine instances interconnected by a commodity ethernet communication fabric. Recently several companies have announced custom cloud environments for the HPC community. Penguin computing has announced their Penguin Computing on Demand (POD) product [12]. POD offers non-virtualized access to the underlying hardware, and supports both gigabit ethernet and DDR Infiniband network interconnects. SGI is now offering a

HPC cloud called Cyclone [13]. Like POD, Cyclone offers non-virtualized access to the hardware, and supports high-performance network interconnects. These offerings aim to provide a complete HPC solution including high-bandwidth storage, optimized HPC applications, and professional services. Our paper does not examine these specialized HPC offerings, but instead focuses on the performance of the (currently) more common virtualized cloud environments.

Section II describes the methods used in this study, including the machines used and the benchmarks run. Section III discusses the performance of the benchmarks and compares the IPM profile from the EC2 with that of a science-oriented commodity cluster. Section IV describes the tools used, and the impediments encountered in attempting to benchmark EC2 performance. Section V describes related work, and Section VI offers our conclusions.

II. METHODS

A. Machines Used In Study

All results were obtained during normal, multi-user, production periods on all machines.

1) *Carver*: is a 400 node IBM iDataPlex cluster located at the National Energy Research Scientific Computing Center (NERSC), which is part of Lawrence Berkeley National Laboratory (LBNL). It has quad-core Intel Nehalem processors running at 2.67 GHz, with dual socket nodes and a single Quad Data Rate (QDR) IB link per node to a network that is locally a fat-tree with a global 2D-mesh. Each node has 24 GB of RAM (3 GB per core). All codes compiled on Carver used version 10.0 of the Portland Group suite and 1.4.1 of Open MPI.

2) *Franklin*: is a 9660 node Cray XT4 supercomputer and is also located at NERSC. Each XT4 compute node contains a single quad-core 2.3 GHz AMD Opteron "Budapest" processor, which is tightly integrated to the XT4 interconnect via a Cray SeaStar-2 ASIC through a 6.4 GB/s bidirectional HyperTransport interface. All the SeaStar routing chips are interconnected in a 3D torus topology, where each node has a direct link to its six nearest neighbors. Each node has 8 GB of RAM (2 GB per core). Codes were compiled with the Pathscale (MAESTRO) and Portland Group version 9.0.4 (all others) compilers.

3) *Lawrencium*: is a 198-node (1584 core) Linux cluster operated by the Information Technology Division at LBNL. Each compute node is a Dell Poweredge 1950 server equipped with two Intel Xeon quad-core 64 bit, 2.66GHz Harpertown processors, connected to a Dual Data Rate (DDR) Infiniband network configured as a fat tree with a 3:1 blocking factor. Each node contains 16 GB of RAM (2 GB per core). Codes were compiled using Intel 10.0.018 and Open MPI 1.3.3.

4) *Amazon EC2*: is a virtual computing environment that provides a web services API for launching and managing virtual machine instances. Amazon provides a number of different instance types that have varying performance characteristics. CPU capacity is defined in terms of an abstract Amazon EC Compute Unit. One EC2 Compute Unit is approximately

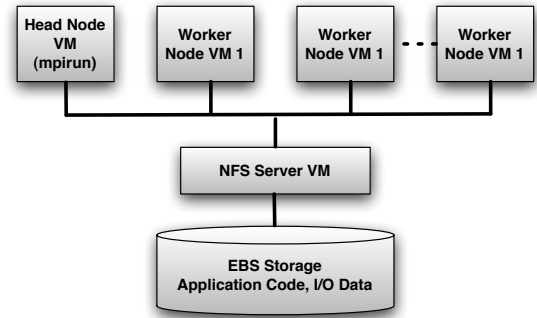


Fig. 1. Virtual Cluster Architecture

equivalent to a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. For our tests we used the m1.large and m1.xlarge instances. The m1.large instance type has four EC2 Compute Units, two virtual cores with two EC2 Compute Units each, and 7.5 GB of memory. The m1.xlarge instance type has eight EC2 Compute Units, four virtual cores with two EC2 Compute Units each, and 15GB of memory. The nodes are connected with gigabit ethernet. To ensure consistency, the binaries compiled on Lawrencium were used on EC2.

All of the nodes are located in the US East region in the same availability zone. Amazon offers no guarantees on proximity of nodes allocated together, and there is significant variability in latency between nodes.

In addition to the variability in network latency, we also see variability in the underlying hardware the virtual machines are running on. By examining `/proc/cpuinfo` we are able to identify the actual CPU type of the un-virtualized hardware. In our test runs, we identified three different CPU's: the Intel Xeon E5430 2.66GHz quad-core processor, the AMD Opteron 270 2.0GHz dual-core processor, and the AMD Opteron 2218 HE 2.6GHz dual-core processor. We have no control over which underlying hardware our virtual machines are instantiated on. Thus, we almost always end up with a virtual cluster running on a heterogeneous set of processors. This heterogeneity also meant we were unable to use any of the processor specific compiler options.

Before we could begin our study, we had to address the major differences between the Amazon Web Services environment and that at a typical supercomputing center. For example, almost all HPC applications assume the presence of a shared parallel filesystem between compute nodes, and a head node that can submit MPI jobs to all of the worker nodes. Running these applications in the cloud requires either that the features of a typical HPC environment are replicated in the cloud, or that the application is changed to accommodate the default configuration of the cloud.

For this paper we chose to replicate a typical HPC cluster environment in the cloud by creating virtual clusters [14], [15]. We used a series of Python scripts to configure a file server, a head node, and a series of worker nodes. The head node could submit MPI jobs to all of the worker nodes, and the file server

provided a shared filesystem between the nodes. This setup is illustrated in Figure 1.

To implement the shared filesystem, we attached an Amazon Elastic Block Store (EBS) [16] device to the fileserver virtual machine. EBS provides a block level storage volume to EC2 instances that persists independently from the instance lifetimes. On top of the EBS volume we built a standard Linux ext3 file system, that was then exported via NFS to all of the virtual cluster nodes.

B. Applications Used In Study

A supercomputer center such as NERSC typically serves a diverse user community. In NERSC's case the community contains over 3,000 users, 400 distinct projects and is comprised of some 600 codes that serve the diverse science needs of the DOE Office of Science research community. Abstracting the salient performance-critical features of such a workload is a challenging task. However, significant workload characterization efforts have resulted in a set of full application benchmarks that span a range of science domains, parallelization schemes, and concurrencies, as well as machine-based characteristics that influence performance such as message size, memory access pattern, and working set sizes. These applications form the basis for the Sustained System Performance (SSP) metric, which better represents the effectiveness of a system for delivered performance on applications rather than peak FLOP rates [17].

As well as being representative of the DOE Office of Science workload, some of these applications have also been used by other federal agencies as they represent significant parts of their workload. MILC and PARATEC were used by the NSF, CAM by NCAR and GAMESS by the NSF and DoD HPCMO. The representation of the methods embodied in our benchmark suite goes well beyond the particular codes employed. For example, PARATEC is representative of methods that constitute one of the largest consumer of supercomputing cycles in computer centers around the world [3]. Therefore, although our benchmark suite was developed with the NERSC workload in mind, we are confident that it is broadly representative of the workloads of many supercomputing centers today. More details about these applications and their computational characteristics can be found in Ref. [18] (as well as in the references cited here).

The typical problem configurations for these benchmarks are defined for much larger "capability" systems, so we had to construct reduced size problem configurations to target the requirements of mid-range workloads that are the subject of this study. For example, many of the input configurations were constructed for a system acquisition (begun during 2008) that resulted in a 1-PetaFlop peak resource that will contain over 150,000 cores - considerably larger than it is possible to run in today's commercial cloud infrastructures. Thus, problem sets were modified to use smaller grids and concomitant concurrencies along with shorter iteration spaces and/or shorter simulation durations. Additionally we also modified the problem

configurations to eliminate any significant I/O because I/O performance is beyond the scope of this work.

Next we describe each of the applications that make up our benchmarking suite, describe the parameters they were run with, and comment upon their computation and communication characteristics.

1) *CAM*: The Community Atmosphere Model (CAM) is the atmospheric component of the Community Climate System Model (CCSM) developed at NCAR and elsewhere for the weather and climate research communities [19], [20]. In this work we use CAM v3.1 with a finite volume (FV) dynamical core and a "D" grid (about 0.5 degree resolution). In this case we used 120 MPI tasks and ran for 3 days simulated time (144 timesteps).

CAM uses a formalism effectively containing two different, two-dimensional domain decompositions, one for the dynamics that is decomposed over latitude and vertical level and the other for remapping that is decomposed over longitude-latitude. Optimized transposes move data from the program structures between these decompositions. CAM is characterized by relatively low computational intensity that stresses on-node/processor data movement and relatively long MPI messages that stress interconnect point-to-point bandwidth.

2) *GameSS*: The GAMESS (General Atomic and Molecular Electronic Structure System) code from the Gordon research group at the Department of Energy Ames Lab at Iowa State University contains various important tools for *ab-initio* quantum chemistry calculations. The benchmark used here calculates the B3LYP DFT energy and gradient for a 43 atom molecule and runs on 64 cores. GameSS is the only benchmark for which no problem size scaling was performed.

GAMESS uses an SPMD approach but includes its own underlying communication library, called the Distributed Data Interface (DDI), to present the abstraction of a global shared memory with one-side data transfers even on systems with physically distributed memory. On the cluster systems included here GAMESS was run using socket communication. On the XT4 an MPI implementation of DDI is used in which only one-half of the processors allocated compute while the other half are essentially data movers. GAMESS is characterized by considerable stride-1 memory access - which stresses memory bandwidth - and interconnect collective performance.

3) *GTC*: GTC is a fully self-consistent, gyrokinetic 3-D Particle-in-cell (PIC) code with a non-spectral Poisson solver [21]. It uses a grid that follows the field lines as they twist around a toroidal geometry representing a magnetically confined toroidal fusion plasma. The version of GTC used here uses a fixed, 1-D domain decomposition with 64 domains and 64 MPI tasks. The benchmark runs are for 250 timesteps using 10 particles per grid cell (2 million grid points, 20 million particles). Communications at this concurrency are dominated by nearest neighbor exchange that are bandwidth-bound. The most computationally intensive parts of GTC involve gather/deposition of charge on the grid and particle "push" steps. The charge deposition utilizes indirect addressing and therefore stresses random access to memory.

TABLE I
HPCC PERFORMANCE

Machine	DGEMM Gflops	STREAM GB/s	Latency μ s	Bandwidth GB/s	RandRing Lat. μ s	RandRing BW GB/s	HPL Tflops	FFTE Gflops	PTRANS GB/s	RandAccess GUP/s
Carver	10.2	4.4	2.1	3.4	4.7	0.30	0.56	21.99	9.35	0.044
Franklin	8.4	2.30	7.8	1.6	19.6	0.19	0.47	14.24	2.63	0.061
Lawrencium	9.6	0.70	4.1	1.2	153.3	0.12	0.46	9.12	1.34	0.013
EC2	4.6	1.7	145	0.06	2065.2	0.01	0.07	1.09	0.29	0.004

4) *IMPACT-T*: *IMPACT-T* (Integrated Map and Particle Accelerator Tracking Time) is an object-oriented Fortran90 code from a suite of computational tools for the prediction and performance enhancement of accelerators. It includes the arbitrary overlap of fields from a comprehensive set of beamline elements, and uses a parallel, relativistic PIC method with a spectral integrated Green function solver. A two-dimensional domain decomposition in the y-z directions is used along with a dynamic load balancing scheme based on domain. Hockneys FFT algorithm is used to solve Poissons equation with open boundary conditions. The problems chosen here are scaled down quite considerably from the official NERSC benchmarks, in terms of number of particles and grid size (4X), 2-D processor configuration (64 cores instead of 256 and 1,024), and number of time steps run (100 instead of 2,000). *IMPACT-T* performance is typically sensitive to memory bandwidth and MPI collective performance. (Note that although both GTC and *IMPACT-T* are PIC codes, their performance characteristics are quite different.)

5) *MAESTRO*: *MAESTRO* is used for simulating astro-physical flows such as those leading up to ignition in Type Ia supernovae. Its integration scheme is embedded in an adaptive mesh refinement algorithm based on a hierarchical system of rectangular non-overlapping grid patches at multiple levels with different resolution; however, in this benchmark using *MAESTRO* the grid does not adapt. A multigrid solver is used. Parallelization is via a 3-D domain decomposition in which data and work are apportioned using a coarse-grained distribution strategy to balance the load and minimize communication costs. The *MAESTRO* communication topology pattern is quite unusual and tends to stress simple topology interconnects. With a very low computational intensity the code stresses memory performance, especially latency; its implicit solver technology stresses global communications; and its message passing utilizes a wide range of message sizes from short to relatively moderate. The problem used was the NERSC-6 "Medium" case (512 X 512 X 1024 grid) on 256 cores but for only 3 timesteps. This problem is more typically benchmarked on 512 cores for 10 timesteps.

6) *MILC*: This code represents Lattice Computation that is used to study Quantum ChromoDynamics (QCD), the theory of the sub-atomic "strong" interactions responsible for binding quarks into protons and neutrons and holding them together in the nucleus. QCD discretizes space and evaluates field variables on sites and links of a regular hypercube lattice in four-dimensional space time. It involves integrating an equation of motion for hundreds or thousands of time

steps that requires inverting a large, sparse matrix at each integration step. The sparse, nearly-singular matrix problem is solved using a conjugate gradient (CG) method and many CG iterations are required for convergence. Within a processor, the four-dimensional nature of the problem requires gathers from widely separated locations in memory. The inversion by CG requires repeated three-dimensional complex matrix-vector multiplications, which reduces to a dot product of three pairs of three-dimensional complex vectors. Each dot product consists of five multiply-add operations and one multiply. The parallel programming model for MILC is a 4-D domain decomposition in which each task exchanges data with its eight nearest neighbors as well as participating in the all-reduce calls with very small payload as part of the CG algorithm. MILC is extremely dependent on memory bandwidth and prefetching and exhibits a high computational intensity.

In this work we use a $32 \times 32 \times 16 \times 18$ global lattice on 64 cores with 2 quark flavors, four trajectories and eight steps per trajectory; this results in over 35,000 CG iterations per run. MILC benchmarking at NERSC uses up to 8,192 cores on a $64^3 \times 144$ grid with 15 steps per trajectory.

7) *Paratec*: *PARATEC* (PARALLEL Total Energy Code) performs *ab initio* Density Functional Theory quantum-mechanical total energy calculations using pseudo-potentials, a plane wave basis set and an all-band (unconstrained) conjugate gradient (CG) approach. Part of the calculation is carried out in Fourier space; custom parallel three-dimensional FFTs are used to transform the wavefunctions between real and Fourier space.

PARATEC uses MPI and parallelizes over grid points, thereby achieving a fine-grain level of parallelism. The real-space data layout of wave-functions is on a standard Cartesian grid. In general, the speed of the FFT dominates the runtime, since it stresses global communications bandwidth, though mostly point-to-point, using relatively short messages. Optimized system libraries (such Intel MKL or AMD ACML) are used for both BLAS3 and 1-D FFT; this results in high cache reuse and a high percentage of per-processor peak performance.

The benchmark used here is based on the NERSC-5 input that does not allow any aggregation of the transpose data. The input contains 250 Silicon atoms in a diamond lattice configuration and runs for 6 conjugate gradient iterations. More typically NERSC uses a 686-atom system with 20 conjugate gradient iterations run on 1024 cores for benchmarking. A real science run might use 60 or more iterations.

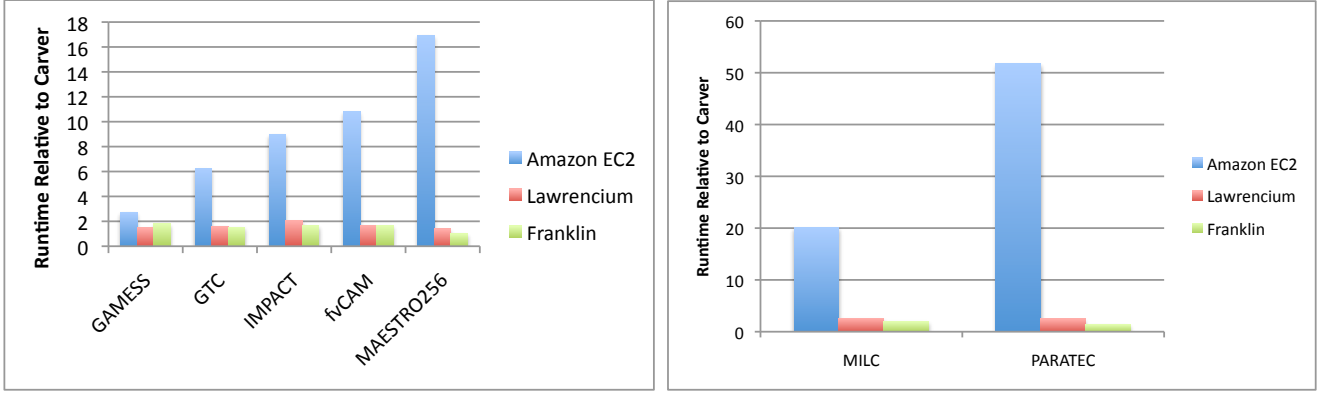


Fig. 2. Runtime of each application on EC2, Lawrencium and Franklin relative to Carver.

8) *HPCC*: In addition to the application benchmarks discussed above, we also ran the High Performance Computing Challenge (HPCC) benchmark suite [22]. HPCC consists of seven synthetic benchmarks: three targeted and four complex. The targeted synthetics are DGEMM, STREAM, and two measures of network latency and bandwidth. These are micro-kernels which quantify basic system parameters that separately characterize computation and communication performance. The complex synthetics are HPL, FFTE, PTRANS, and RandomAccess. These combine computation and communication and can be thought of as very simple proxy applications. Taken together these benchmarks allow for the measurement of a variety of lower-level factors that are important for performance, which is why we chose to use them for this work.

C. Evaluation Methodology

At NERSC timing results from these application benchmarks are used to compute the Sustained System Performance (SSP) metric [17], an aggregate measure of the workload-specific *delivered* performance of a computing system. The SSP is derived from an application performance figure, P_i , expressed in units of GFlops per second per core. Given a system configured with N computational cores, the SSP is the geometric mean of P_i over all M applications, multiplied by N , which is the size of the system being considered.

$$SSP = N \left(\prod_{i=1}^M P_i \right)^{(1/M)}. \quad (1)$$

The floating-point operation count used in calculating P_i for each of the seven component applications has been pre-determined using a hardware performance counter on a single reference system at NERSC, the Cray XT4. The reference counts are combined with the times from other systems to calculate the SSP for those systems.

The SSP is evaluated at discrete points in time and also as an integrated value to give the systems potency, meaning an estimate of how well the system will perform the expected work over some time period. Taken together, the NERSC

SSP benchmarks, their derived aggregate measures, and the entire NERSC workload-driven evaluation methodology create a strong connection between science requirements, how the machines are used, and the tests we use.

III. RESULTS

A. HPC Challenge

The results of running HPCC v.1.4.0 on 64 cores of the four machines in our study are shown in Table I. The DGEMM results are as one would expect based on the properties of the CPUs. The STREAM results show that EC2 is significantly faster for this benchmark than Lawrencium. We believe this is because of the particular processor distribution we received for our EC2 nodes for this test. We had 26 AMD Opteron 270's, 16 AMD Opteron 2218 HE's, and 14 Intel Xeon E5430's, of which this measurement represents an average. The AMD Opteron based systems are known to have better memory performance than the Intel Harpertown systems used in Lawrencium. Both EC2 and Lawrencium are significantly slower than the Nehalem-based Carver system, however.

The network latency and bandwidth results clearly show the difference between the interconnects on the tested systems. For display we have chosen both the average ping-pong latency and bandwidth, and the randomly-ordered ring latency and bandwidth. The ping-pong results show the latency and the bandwidth with no self-induced contention, while the randomly ordered ring tests show the performance degradation with self-contention. The uncontended latency and bandwidth measurements of the EC2 gigabit ethernet interconnect are more than 20 times worse than the slowest other machine. Both EC2 and Lawrencium suffer a significant performance degradation when self-contention is introduced. The EC2 latency is 13 times worse than Lawrencium, and more than 400 times slower than a modern system like Carver. The bandwidth numbers show similar trends: EC2 is 12 times slower than Lawrencium, and 30 times slower than Carver.

We now turn our attention to the complex synthetics. The performance of these is sensitive to characteristics of both the

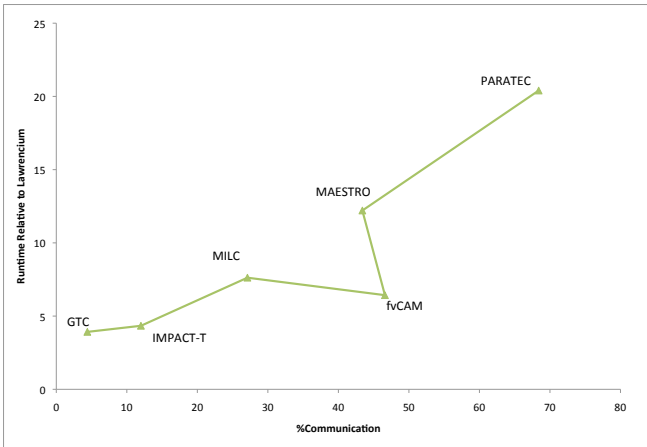


Fig. 3. Correlation between the runtime of each application on EC2 and the amount of time an application spends communicating.

processor and the network, and their performance gives us some insight into how real applications may perform on EC2.

HPL is the high-performance version of the widely-reported Linpack benchmark, which is used to determine the TOP500 list. It solves a dense linear system of equations and its performance depends upon DGEMM and the network bandwidth and latency. On a typical high performance computing system today roughly 90% of the time is spent in DGEMM and the results for the three HPC systems illustrate this clearly. However, for EC2 the less capable network clearly inhibits overall HPL performance, by a factor of six or more. The FFTE benchmark measures the floating point rate of execution of a double precision complex one-dimensional discrete Fourier transform, and the PTRANS benchmark measures the time to transpose a large matrix. Both of these benchmarks performance depends upon the memory and network bandwidth and therefore show similar trends. EC2 is approximately 20 times slower than Carver and four times slower than Lawrencium in both cases. The RandomAccess benchmark measures the rate of random updates of memory and its performance depends on memory and network latency. In this case EC2 is approximately 10 times slower than Carver and three times slower than Lawrencium.

Overall the results of the HPCC runs indicate that the lower performing network interconnect in EC2 has a significant impact upon the performance of even very simple application proxies. This is illustrated clearly by the HPL results which are significantly worse than would be expected from simply looking at the DGEMM performance.

B. Applications

Figure 2 shows the relative runtime of each of our test applications relative to Carver, which is the newest, and therefore fastest, machine in our testbed. For these applications, at these concurrencies, Franklin and Lawrencium are between $1.4\times$ and $2.6\times$ slower than Carver. For EC2 the range of performance observed is significantly greater. In the best case, GAMESS, EC2 is only $2.7\times$ slower than Carver. For the

worst case, PARATEC, EC2 is more than $50\times$ slower than Carver. This large spread of performance simply reflects the different demands each application places upon the network, as in the case of the compact applications that were described in the previous section. Qualitatively we can understand the differences in terms of the performance characteristics of each of the applications described in Section II-B. PARATEC shows the worst performance on EC2, $52\times$ slower than Carver. It performs 3-DFFT's, and the global (i.e., all-to-all) data transposes within these FFT operations can incur a large communications overhead. MILC ($20\times$) and MAESTRO ($17\times$) also stress global communication, but to a lesser extent than PARATEC. CAM ($11\times$), IMPACT ($9\times$) and GTC ($6\times$) are all characterized by large point-to-point communications, which do not induce quite as much contention as global communication, hence their performance is not impacted quite as much. GAMESS ($2.7\times$), for this benchmark problem, places relatively little demand upon the network, and therefore is hardly slowed down at all on EC2.

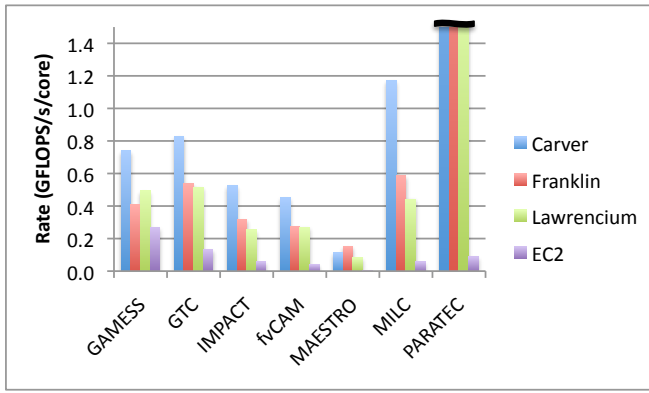
Qualitatively, it seems that those applications that perform the most collective communication with the most messages are those that perform the worst on EC2. To gain a more quantitative understanding, we perform a more detailed analysis, which is described in the next section.

C. Performance Analysis Using IPM

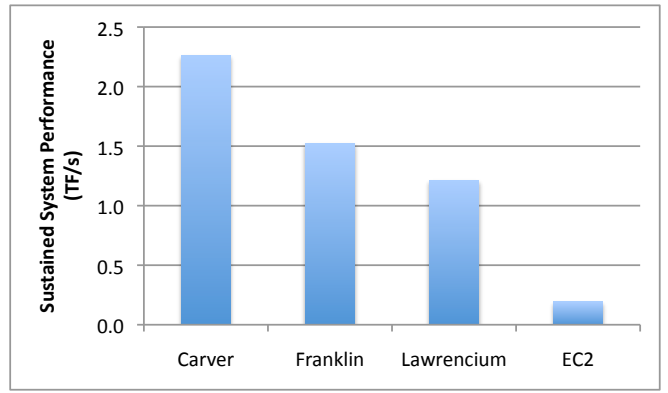
To understand more deeply the reasons for the poor performance of EC2 in comparison to the other platforms we performed additional experiments using the Integrated Performance Monitoring (IPM) framework [11], [23]. IPM is a profiling tool that uses the MPI profiling interface to measure the time taken by an application in MPI on a task-by-task basis. This allows us to examine the relative amounts of time taken by an application for computing and communicating, as well as the types of MPI calls made. These measurements will enable us to determine which particular aspects of the EC2 hardware configuration are most inhibiting performance on an application by application basis in a quantitative manner. Previous measurements with IPM have shown that it has extremely low overhead [23], less than 2%, giving us confidence that by instrumenting the applications with IPM we are not altering their runtime characteristics.

One of the simplest metrics available from IPM is the percentage of the runtime that the application spends communicating (time in the MPI library to be precise). Figure 3 shows the relative runtime on EC2 compared to Lawrencium plotted against the percentage communication for each application as measured on Lawrencium. The overall trend is clear: the greater the fraction of its runtime an application spends communicating, the worse the performance is on EC2.

The principle exception to this trend is fvCAM, the performance on EC2 is much faster than would be expected from the simple considerations described above. Currently we do not have a quantitative explanation for this observation, but given the considerable variability EC2 performance we have observed it seems most likely that this is simply a judiciously



(a) Sustained performance per core for each of the applications on each of the machines. The PARATEC values for Carver, Franklin and Lawrencium are truncated for clarity. They are 4.6, 3.2 and 1.8 Gflop/s/core respectively.



(b) Sustained system performance for Carver, Franklin, Lawrencium and EC2 for the benchmark suite defined in Section II-B.

Fig. 4. Sustained performance a) on a per application and per machine basis b) on a per machine basis only.

fast run. Furthermore the magnitude of the discrepancy is also consistent with this hypothesis. Additional investigations are underway.

Thus using quantitative analysis based upon instrumenting our applications with IPM we have explained the reason that each application is slowed down by different amounts when running on EC2.

D. Sustained System Performance

Figure 4a shows the sustained floating point rate per core achieved by each application on each machine. Often this is presented as percentage of peak, but in this case the inhomogeneous CPU environment on EC2 makes that impossible to calculate. The rate achieved by the applications is representative of their computational intensity (ratio of floating point operations to memory operations) as well as the amount of time they spend communicating.

In order to combine these measurements into a single metric that represents the overall performance for this workload we calculate the SSP for these applications on a 3200 core machine. ($N = 3200$ in Equation 1.) We chose 3200 cores as it is the size of Carver, and represents a reasonable mid-size HPC system today. (Ideally of course our calculation would be based upon the size of EC2 as that is what we are evaluating here; unfortunately that is not possible.) We note that the particular number of cores used only affects the magnitude of the SSP, not the relative ratios between values for different machines.

The results of the SSP calculation are shown in Figure 4b. It shows that the sustained performance for Carver is more than 4 TF/s or almost 7% of peak. Franklin, Lawrencium and EC2 are 1.5, 1.2 and 0.2 TF/s respectively. (Note that this value differs from the published one for Franklin because, as described in Section II-B, the problem sizes in this work are drastically reduced.) Thus for this workload in aggregate the slowest of our HPC systems, Lawrencium, is six times faster than EC2, and the fastest, Carver is twenty times faster.

IV. DISCUSSION

As we saw in the last section, the overall performance of EC2, running this workload, is significantly slower than a typical mid-size cluster like Lawrencium. Our experiments show that this is largely because of the network interconnect available in EC2. This is borne out by the data in Section III-C, showing that the larger percentage of time an application spends in communication, the worse its overall EC2 performance will be. In addition, the applications communication pattern affects how it uses the network interconnect, and will also affect the performance. As seen with PARATEC, all-to-all communications performed to transpose data can severely hamper performance.

While conducting this study, we discovered a significant amount of variability in application performance. One of the sources of this variability in the Amazon cloud environment is the non-homogeneous nature of the systems allocated, as described in Section II-A4. During our testing we saw three different processor types: two AMD Opteron CPUs and one Intel Xeon, with the particular distribution varying from test to test. This heterogeneity makes benchmarking difficult, as it is hard to compare two different runs as a completely different set of processors may be acquired on the next test. In this work our benchmarks are a snapshot of what the performance was with a particular set of resources at a particular time. We note that for application developers this inhomogeneity causes difficulties with performance tuning and load-balancing applications. Another source of variability is introduced by network contention. The switching fabric is shared between all of the EC2 users, and may be heavily contended for.

One last source of variability over which we have no control, nor way of discovering, is if we are sharing the un-virtualized hardware or not. For example, multiple virtual machines may be running on the same physical machine. If one of our nodes is sharing the hardware with another virtual machine that is making extensive use of some hardware component, like the network interface, our performance will suffer.

To study this variability, and understand quantitatively its

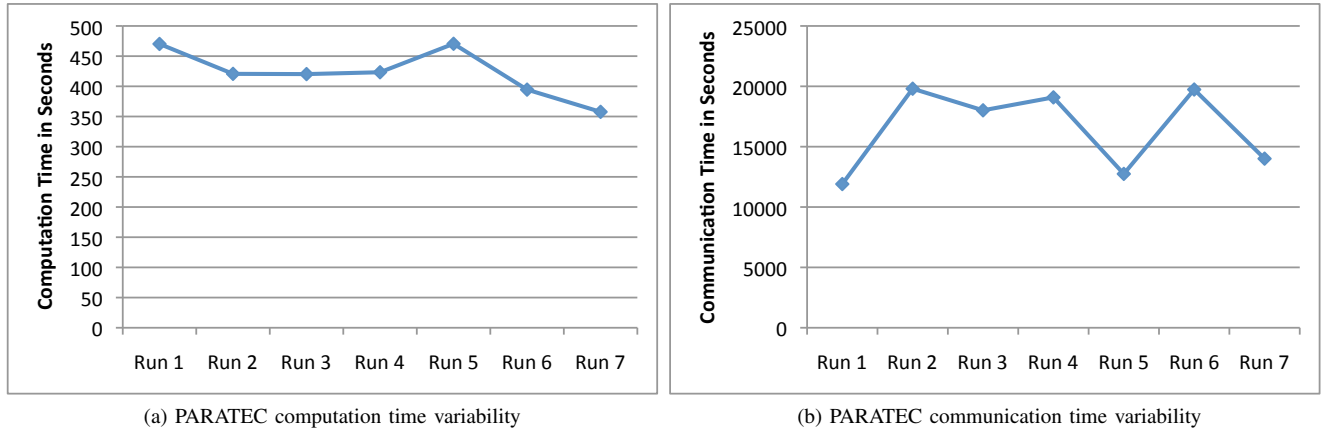


Fig. 5. Paratec performance variability on EC2.

effect, we conducted seven runs of the PARATEC application using different virtual cluster instances. In Figure 5a we show the time spent in computation for each of these runs as measured using IPM. Overall there is a 30% variability seen in compute time which can be explained by examining the processor distribution acquired for each run. In the first run, which was the slowest, our virtual cluster had 48 of the 2.0GHz AMD 270 processors, and only 2 of the 2.66GHz Intel E5430 processors. On the other hand, run seven spent the least amount of time in computation. For that run we had 18 of the slower AMD 270's, and 26 of the Intel E5430's, and 8 of the 2.66GHz AMD 2218 HE processors. As we anticipated, computation time can vary significantly based on the distribution of processors acquired. In Figure 5b we show the time spent in communication for these runs. As discussed in Section III-B, PARATEC's communication pattern performs particularly poorly on EC2. Communication accounts for approximately 97% of the overall runtime, and accounts for most of the variability in runtime. The difference between the maximum and minimum runtime is 7,900 seconds, or approximately 42% of the mean runtime. This clearly shows the extreme variability in network performance within EC2.

In an attempt to mitigate the effect of the poor performance of the EC2 network (and its variability) we also performed experiments using fewer overall MPI tasks, for the PARATEC and MILC applications on both EC2 and Lawrencium. The results of these are shown in Figures 6a and 6b. The principle observation is that even though we are now using one-half or one-quarter as many nodes the EC2 runtime still shows significant effects due to variability. In fact these are completely dominant, suggesting that unless one is using a loosely coupled application, one is better off running applications exclusively within a single EC2 node. The Lawrencium results show the expected strong-scaling behavior.

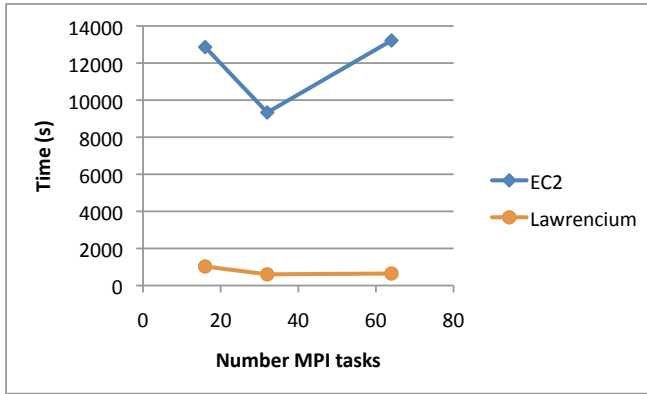
We don't explicitly address cost in this work, because it is highly dependent on the specific application and operational requirements of a given collaboration. However there is a direct correlation between cost and performance, and we believe that these performance metrics will prove to be an

invaluable tool for computing the true cost of running on the cloud for a given scientific group. This cost will depend on several factors including the size of the application, its need for concurrency, its IO requirements, its fault-tolerance and the general software integration and porting challenges. These are highly site and application dependent.

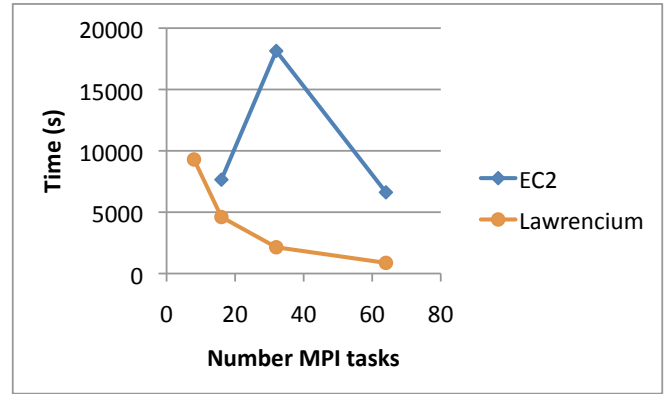
We also do not address I/O performance in this paper. While we recognize that I/O performance is critical to many HPC applications, we chose to focus this study on computational performance. We expect that future work will examine I/O and Wide Area Network (WAN) performance from a scientific application perspective.

One major lesson of this study was that the mean time between failures (MTBF) of individual nodes in a virtual cluster is significantly higher than in a traditional HPC environment. Traditional MPI based applications are intolerant of node failures and other transient errors. Our experience with the Amazon Web Services environment is that a variety of transient failures can occur, including an inability to access the user-data passed in during image startup, failure to properly configure the network, failure to boot properly, and other performance perturbations, including intermittent virtual machine hangs. While none of these errors occurred frequently, they do in aggregate happen often enough that it becomes a significant barrier to running MPI applications. Approximately one in ten runs would need to be restarted due to some failure.

A common failure that must be handled by the virtual cluster software is resource unavailability. In general, we found that the software creating the virtual cluster cannot assume that it will always acquire all of the requested resources. Allocating 128 or more cores at once is not always practical, and results in indefinite hangs and costly idling of resources if the request cannot be fulfilled. For this reason, we confined our tests to running on no more than 256 cores. Scheduling resources for larger core counts appears to be impractical at this time without moving to higher-cost reservation services, thus severely limiting the ability to run many of the applications common at supercomputing centers.



(a) PARATEC runtime scaling on EC2 and Lawrencium.



(b) MILC runtime scaling on EC2 and Lawrencium

Fig. 6. Paratec and MILC scaling behavior on Lawrencium and EC2.

V. RELATED WORK

A number of different groups have conducted feasibility studies of running their scientific applications in the Amazon cloud. In addition, previous work has examined the performance of individual Amazon AWS components, e.g., the simple storage service (S3) [24].

Hazelhurst examines the performance of the bioinformatics application WCD [25]. The performance and storage costs of running the Montage workflow on EC2 are detailed by Deelman et. al. [26]. The High-Energy and Nuclear Physics (HENP) STAR experiment has examined the costs and challenges associated with running their analysis application on the EC2 cloud [15], [27], [28]. In previous work we examined the usefulness of cloud computing for e-Science applications [29], [30].

Standard benchmarks have also been evaluated on Amazon EC2. Napper et. al. examine the performance of the Linpack benchmarks on different EC2 instance types [31]. The NAS benchmarks have been run by Evangelinos et. al. [32] and Masud [33]. Osterman et. al. ran a variety of microbenchmarks and kernels [34].

This work is unique in examining the performance of a set of applications that represent the typical workload run at a major supercomputing center. The applications chosen represent both the range of science done and the algorithms typical of supercomputing codes. More importantly, by analyzing the running code using IPM we are able to profile the underlying characteristics of the application, and can quantitatively identify the major performance bottlenecks and resource constraints with respect to the EC2 cloud.

VI. CONCLUSIONS

While cloud computing has proven itself useful for a wide range of e-Science applications, its utility for more tightly-coupled HPC applications has not been proven. In this paper we have quantitatively examined the performance of a set of benchmarks designed to represent a typical HPC workload run on Amazon EC2. Our data clearly shows a strong correlation between the percentage of time an application spends

communicating, and its overall performance on EC2. The more communication, the worse the performance becomes. We were also able to see that the communication pattern of the application can have a significant impact on performance. Applications, like PARATEC, with significant global communication perform relatively worse than those with less global communication. Finally we learned that the amount of variability in EC2 performance can be significant. Variability is introduced by the shared nature of the virtualized environment, by the network, and by differences in the underlying non-virtualized hardware.

VII. ACKNOWLEDGEMENTS

This work was funded in part by the Advanced Scientific Computing Research (ASCR) in the DOE Office of Science under contract number DE-C02-05CH11231. NJW was supported by the NSF under award OCI-0721397. This research used resources of the National Energy Research Scientific Computing Center, under Contract No. DE-AC02-05CH11231. The authors would like to thank CITRIS, UC Berkeley for Amazon EC2 access.

REFERENCES

- [1] "MidRange Computing in Support of Science at Office of Science Laboratories," <http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/MidrangeReportFinal.pdf>, DOE ASCR, Tech. Rep., 2009.
- [2] "Amazon Elastic Compute Cloud," <http://aws.amazon.com/ec2/>.
- [3] L. Oliker, A. Canning, J. Carter, J. Shalf, and S. Ethier, "Scientific computations on modern parallel vector systems," in *Proc. SC04: International Conference for High Performance Computing, Networking, Storage and Analysis*, Pittsburgh, PA, Nov6-12, 2004.
- [4] L. Oliker, J. Carter, M. Wehner et al., "Leading computational methods on scalar and vector HEC platforms," in *Proc. SC05: International Conference for High Performance Computing, Networking, Storage and Analysis*, Seattle, WA, Nov 12-18, 2005.
- [5] J. Carter, L. Oliker, and J. Shalf, "Performance evaluation of scientific applications on modern parallel vector systems," in *VECPAR: High Performance Computing for Computational Science*, Rio de Janeiro, Brazil, July 10-12, 2006.
- [6] L. Oliker, A. Canning, J. Carter et al., "Scientific application performance on candidate petascale platforms," in *Proc. IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Long Beach, CA, Mar 26-30, 2007.

- [7] T. H. Dunigan Jr., J. S. Vetter, J. B. White III, and P. H. Worley, "Performance evaluation of the Cray X1 distributed shared-memory architecture," *IEEE Micro*, vol. 25(1), pp. 30–40, Jan/Feb 2005.
- [8] K. Nakajima, "Three-level hybrid vs. flat mpi on the earth simulator: Parallel iterative solvers for finite-element method," in *Proc. 6th IMACS Symposium Iterative Methods in Scientific Computing*, vol. 6, Denver, CO, Mar 27–30, 2003.
- [9] J. Vetter, S. Alam, T. Dunigan, Jr. *et al.*, "Early evaluation of the Cray XT3," in *Proc. IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Rhodes Island, Greece, April 25–29, 2006.
- [10] F. Gygi, E. W. Draeger, B. R. de Supinski *et al.*, "Large-scale first-principles molecular dynamics simulations on the BlueGene/L platform using the Qbox code," in *Proc. SC05: International Conference for High Performance Computing, Networking, Storage and Analysis*, Seattle, WA, Nov 12–18, 2005.
- [11] D. Skinner, "Integrated Performance Monitoring: A portable profiling infrastructure for parallel applications," in *Proc. ISC2005: International Supercomputing Conference*, Heidelberg, Germany, 2005.
- [12] "Penguin computing on demand," http://www.penguincomputing.com/POD/Penguin_On_Demand.
- [13] "Sgi cyclone," http://www.sgi.com/products/hpc_cloud/cyclone/.
- [14] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, and X. Zhang, "Virtual clusters for grid communities," in *Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid*. Citeseer, 2006, pp. 513–520.
- [15] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, and M. Tsugawa, "Science clouds: Early experiences in cloud computing for scientific applications," *Cloud Computing and Applications*, vol. 2008, 2008.
- [16] "Amazon Elastic Block Store," <http://aws.amazon.com/ebs/>.
- [17] W. Kramer, J. Shalf, and E. Strohmaier, "The NERSC Sustained System Performance (SSP) Metric," 2005.
- [18] K. Antypas, J. M. Shalf, and H. Wasserman, "NERSC-6 workload analysis and benchmark selection process," LBNL, Tech. Rep., 2008.
- [19] "CAM3.1," <http://www.cesm.ucar.edu/models/atm-cam/>.
- [20] "Community Atmosphere Model," <http://www.cgd.ucar.edu/csm/models/atm-cam>.
- [21] W. W. Lee, "Gyrokinetic particle simulation model," *J. Comp. Phys.*, vol. 72, 1987.
- [22] "HPCC benchmark web page: <http://icl.cs.utk.edu/hpcc/>."
- [23] N. J. Wright, W. Pfeiffer, and A. Snavely, "Characterizing parallel scaling of scientific applications using IPM," in *The 10th LCI International Conference on High-Performance Clustered Computing*, March 10–12, 2009.
- [24] M. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon S3 for science grids: a viable solution?" in *Proceedings of the 2008 international workshop on Data-aware distributed computing*. ACM, 2008, pp. 55–64.
- [25] S. Hazelhurst, "Scientific computing using virtual high-performance computing: a case study using the Amazon elastic computing cloud," in *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*. ACM, 2008, pp. 94–103.
- [26] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. IEEE Press, 2008, pp. 1–12.
- [27] K. Keahey, T. Freeman, J. Lauret, and D. Olson, "Virtual workspaces for scientific applications," in *Journal of Physics: Conference Series*, vol. 78. Institute of Physics Publishing, 2007, p. 012038.
- [28] K. Keahey, "Cloud Computing for Science," in *Proceedings of the 21st International Conference on Scientific and Statistical Database Management*. Springer-Verlag, 2009, p. 478.
- [29] L. Ramakrishnan, K. R. Jackson, S. Canon, S. Cholia, and J. Shalf, "Defining Future Platform Requirements for e-Science Clouds," in *Proceedings of the ACM Symposium on Cloud Computing (SoCCi)*. ACM, 2010.
- [30] J. Li, D. Agarwal, M. Humphrey, C. van Ingen, K. Jackson, and Y. Ryu, "esience in the cloud: A modis satellite data reprojection and reduction pipeline in the windows azure platform," in *Proceedings of the 24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010)*, Atlanta, GA, April 19–23, 2010.
- [31] J. Napper and P. Bientinesi, "Can cloud computing reach the top500?" in *Proceedings of the combined workshops on UnConventional high performance computing workshop plus memory access workshop*. ACM, 2009, pp. 17–20.
- [32] C. Evangelinos and C. Hill, "Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazons EC2," *ratio*, vol. 2, no. 2.40, pp. 2–34, 2008.
- [33] R. Masud, "High Performance Computing with Clouds."
- [34] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "An early performance analysis of cloud computing services for scientific computing," *Delft University of Technology, Tech. Rep.*, 2008.